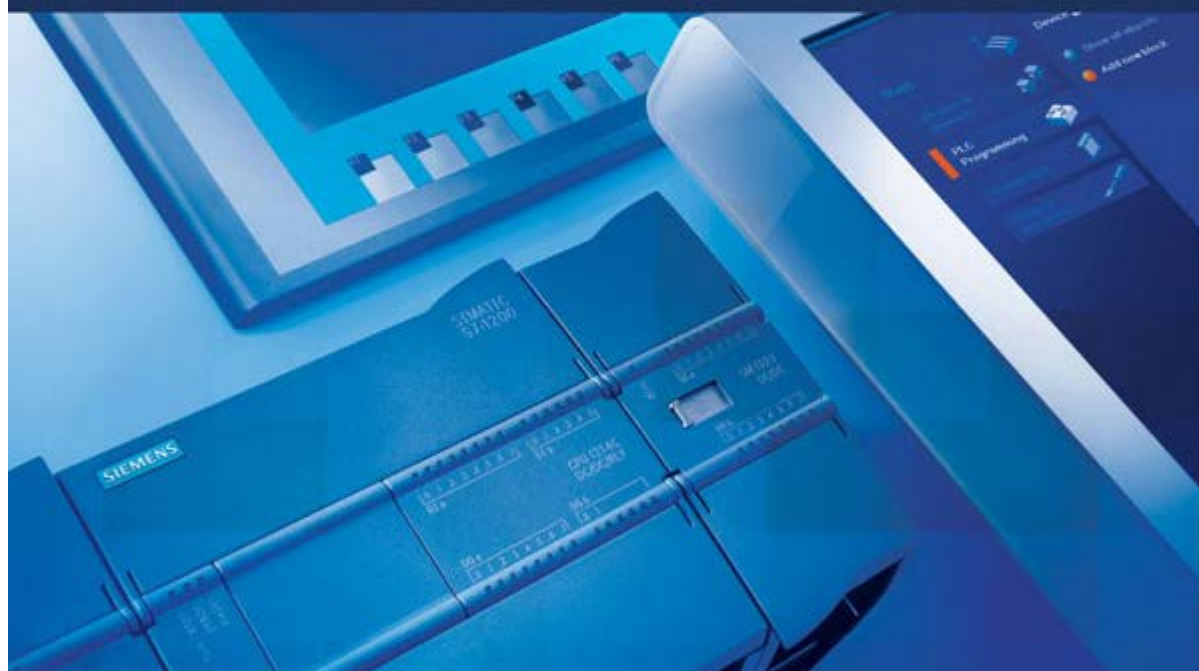


FRANCESCO PRUDENTE

PLC S7-1200

TEORIA E APLICAÇÕES

CURSO INTRODUTÓRIO



PLC S7-1200

TEORIA E APLICAÇÕES

Curso Introdutório



O GEN | Grupo Editorial Nacional reúne as editoras Guanabara Koogan, Santos, Roca, AC Farmacêutica, Forense, Método, LTC, E.P.U. e Forense Universitária, que publicam nas áreas científica, técnica e profissional.

Essas empresas, respeitadas no mercado editorial, construíram catálogos inigualáveis, com obras que têm sido decisivas na formação acadêmica e no aperfeiçoamento de várias gerações de profissionais e de estudantes de Administração, Direito, Enfermagem, Engenharia, Fisioterapia, Medicina, Odontologia, Educação Física e muitas outras ciências, tendo se tornado sinônimo de seriedade e respeito.

Nossa missão é prover o melhor conteúdo científico e distribuí-lo de maneira flexível e conveniente, a preços justos, gerando benefícios e servindo a autores, docentes, livreiros, funcionários, colaboradores e acionistas.

Nosso comportamento ético incondicional e nossa responsabilidade social e ambiental são reforçados pela natureza educacional de nossa atividade, sem comprometer o crescimento contínuo e a rentabilidade do grupo.

PLC S7-1200

TEORIA E APLICAÇÕES

Curso Introdutório

Francesco Prudente

Professor Titular do Laboratório de Eletrotécnica e Automação Industrial
no Istituto di Istruzione Superiore di Stato – IPSIA Marcora – Milão/Itália



O autor e a editora empenharam-se para citar adequadamente e dar o devido crédito a todos os detentores dos direitos autorais de qualquer material utilizado neste livro, dispondo-se a possíveis acertos caso, inadvertidamente, a identificação de algum deles tenha sido omitida.

Não é responsabilidade da editora nem do autor a ocorrência de eventuais perdas ou danos a pessoas ou bens que tenham origem no uso desta publicação.

Apesar dos melhores esforços do autor, do editor e dos revisores, é inevitável que surjam erros no texto. Assim, são bem-vindas as comunicações de usuários sobre correções ou sugestões referentes ao conteúdo ou ao nível pedagógico que auxiliem o aprimoramento de edições futuras. Os comentários dos leitores podem ser encaminhados à **LTC — Livros Técnicos e Científicos Editora** pelo e-mail lrc@grupogen.com.br.

Direitos exclusivos para a língua portuguesa

Copyright © 2014 by

LTC — Livros Técnicos e Científicos Editora Ltda.

Uma editora integrante do GEN | Grupo Editorial Nacional

A presente obra fornece ao usuário ideias e sugestões no uso desse sistema. Não substitui de forma alguma o manual de sistema do dispositivo, que deverá ser obrigatoriamente consultado. A Siemens e o autor não são legalmente responsáveis por eventuais danos ou lesões consequentes ao uso dessas aplicações. Qualquer forma de duplicação ou distribuição, inclusive de algumas partes, é admitida conforme expressa autorização de SIEMENS.

Reservados todos os direitos. É proibida a duplicação ou reprodução deste volume, no todo ou em parte, sob quaisquer formas ou por quaisquer meios (eletrônico, mecânico, gravação, fotocópia, distribuição na internet ou outros), sem permissão expressa da editora.

Travessa do Ouvidor, 11

Rio de Janeiro, RJ – CEP 20040-040

Tels.: 21-3543-0770 / 11-5080-0770

Fax: 21-3543-0896

lrc@grupogen.com.br

www.lrceditora.com.br

Capa: Leônidas Leite

Produção Digital: Geethik

CIP-BRASIL. CATALOGAÇÃO NA PUBLICAÇÃO
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

P966p

Prudente, Francesco

PLC S7-1200 : teoria e aplicações : curso introdutório / Francesco Prudente. - 1. ed. - Rio de

Janeiro : LTC, 2014.

il. ; 24 cm

Inclui bibliografia e índice

ISBN 978-85-216-2569-8

1. Automação Industrial. I. Título.

13-05854

CDD: 629.895

CDU: 62-5

Este livro é voltado aos profissionais que trabalham no setor de automação industrial e têm a intenção de aprofundar seus conhecimentos no campo dos Controladores Lógicos Programáveis (PLC) do fabricante Siemens Automation.

É ainda recomendado para profissionais da indústria e estudantes de cursos técnicos profissionalizantes e universitários.

Os conceitos trabalhados são ilustrados com muitos exemplos concretos e funcionais. Um operador do setor ou um estudante, independentemente do seu nível e de sua experiência, pode compreender sem dificuldade os conceitos e as aplicações.

O tratamento dado aos conceitos e às aplicações não pressupõe experiência anterior e no campo da programação com computador. Todos os exemplos e as aplicações propostos foram rigorosamente comprovados e controlados em laboratório. O único pré-requisito necessário é o conhecimento básico do ambiente Windows no PC.

Neste livro utilizamos como modelo as novas CPUs SIMATIC S7-1200 e o software STEP 7 Basic conforme a norma IEC 61131-3 em ambiente Windows, com o objetivo de fornecer noções básicas do hardware do PLC e de suas principais interfaces de I/O e módulos especiais.

O livro fornece ainda a base teórica e prática para a programação desse dispositivo, o PLC, que já se tornou indispensável em qualquer atividade de automação.

Além disso, o próprio sistema operacional STEP 7 Basic, com o conceito de TIA (Totally Integrated Automation), permite, com uma única plataforma, uma ferramenta completa para configuração, programação, comunicação e visualização de qualquer projeto de automação industrial.

No sistema operacional STEP 7 Basic, o sistema supervisor SIMATIC WinCC Basic já vem integrado para a configuração dos novos painéis operadores da série SIMATIC HMI Basic Panels. A obra é organizada em 11 capítulos didaticamente subdivididos em seções, de forma a permitir uma sequência pedagógica sistemática, com muitos exemplos, figuras e aplicações práticas a serem verificados no transcorrer do texto, elevando progressivamente o grau de dificuldade a cada novo capítulo.

É oportuno assinalar que a codificação dos símbolos gráficos para os esquemas elétricos toma como base a norma internacional IEC 60617-1...13, relativa aos símbolos gráficos para diagramas elétricos.

O Autor

Agradecimento Especial

À professora Mestre Marla Cristiane A. Medeiros, pelas ilustrações e pela tradução dos originais e de apontamentos do autor para o português.

F.P.

Capítulo 1 LÓGICA PROGRAMADA

- 1.0 Definição de *Programmable Logic Controller*
- 1.1 Evolução dos Modernos PLC
- 1.2 Vantagens e Precauções na Utilização do PLC
- 1.3 Confiabilidade e Segurança no Sistema PLC
- 1.4 Significado de Hardware e Software
- 1.5 Descrição do Sistema PLC
- 1.6 Considerações Finais

Capítulo 2 ARQUITETURA DO SISTEMA SIMATIC S7-1200

- 2.0 Generalidades
- 2.1 Os Componentes do Sistema S7-1200

Capítulo 3 HARDWARE DO PLC S7-1200

- 3.0 Generalidades
- 3.1 Módulo CPU S7-1200
 - 3.1.1 Entradas e Saídas I/O — On Board
 - 3.1.2 O Signal Board (SB)
 - 3.1.3 Modo Operacional da CPU
 - 3.1.4 Ciclo de Scan
 - 3.1.5 A SIMATIC Memory Card
- 3.2 Ligação com a Porta PROFINET
 - 3.2.1 Interface Integrada PROFINET
- 3.3 Os Módulos de Comunicação (CM)
- 3.4 Módulo Compact Switch (CSM)
- 3.5 Módulo Power Supply (PM)
- 3.6 Simulador de Entradas
- 3.7 O Signal Module (SM)
 - 3.7.1 Propriedade das Entradas e Saídas Digitais
 - 3.7.2 Propriedade das Entradas e Saídas Analógicas
 - 3.7.3 O Novo Módulo SM 1231 4 × AI 16 Bits
- 3.8 Características Técnicas Principais das CPUs S7-1200
- 3.9 HMI Basic Panels

Capítulo 4 INTRODUÇÃO À LINGUAGEM DE PROGRAMAÇÃO CONFORME NORMA IEC 61131-3

- 4.0 Generalidades

- 4.1 A Norma IEC 61131-3
- 4.2 Linguagem de Programação Textual
- 4.3 Linguagem de Programação Gráfica
- 4.4 Linguagem em Lista de Instruções – IL
- 4.5 Linguagem com Texto Estruturado – ST
- 4.6 Linguagem LADDER – LD
- 4.7 Linguagem com Diagrama de Blocos Funcionais – FBD

Capítulo 5 INTRODUÇÃO AO STEP 7 BASIC

- 5.0 Generalidades
- 5.1 As Novidades da STEP 7 V11.0
- 5.2 Portal View (Visual Portal) e Project View de STEP 7 Basic
- 5.3 Instalação de STEP 7 Basic
- 5.4 Primeiros Passos no Portal View
- 5.5 Guia On-line
- 5.6 Editor de Programação
- 5.7 Editor Tabela dos Símbolos
- 5.8 Criação de um Projeto com STEP 7 Basic
- 5.9 Exemplo de Criação de um Projeto
 - 5.9.1 Configurar a Porta de Comunicação
 - 5.9.2 Ligação dos Dispositivos
 - 5.9.3 Download e Simulação do Programa

Capítulo 6 CARACTERÍSTICAS DO SOFTWARE DO PLC S7-1200

- 6.0 Generalidades
- 6.1 Bit, Byte, Word
- 6.2 Tipos de Dados na CPU S7-1200
- 6.3 Endereçamento dos Dados
- 6.4 Modalidade de Endereçamento das Principais Áreas/Dados do PLC S7-1200
- 6.5 Configuração dos Endereços com STEP 7 Basic
- 6.6 A Programação de Blocos
 - 6.6.1 Tipos de Blocos
 - 6.6.2 Organização dos Blocos OB
- 6.7 Programação Linear e Estruturada
- 6.8 Exemplo de um DB Instance e DB Global
- 6.9 Blocos com Parâmetros
 - 6.9.1 Exemplo de Bloco com Parâmetros
 - 6.9.2 Configuração de um Bloco com Parâmetros

Capítulo 7 INTRODUÇÃO ÀS LINGUAGENS DE PROGRAMAÇÃO LADDER E FBD

- 7.0 Generalidades
- 7.1 O Esquema Funcional Europeu
- 7.2 O Esquema Funcional Norte-americano
- 7.3 A Lógica por Contatos
- 7.4 Símbolos Simples
- 7.5 Ligação Série e Paralelo
- 7.6 A Lógica por Portas Digitais
- 7.7 Transformação de Esquema Funcional em Diagrama Ladder
- 7.8 Introdução às Aplicações
- 7.9 A Função SET/RESET
- 7.10 Aplicação: Relé de Passo

Capítulo 8 DETECÇÃO DE BORDA DE DESCIDA E SUBIDA

- 8.0 Generalidades
- 8.1 Definição de Borda de Descida e Subida
- 8.2 Construção da Operação de Detecção da Borda de Descida e Subida
- 8.3 Operação Especial de Detecção com a CPU S7-1200
- 8.4 Aplicação: Partida e Parada com Segurança de um MAT (Motor Assíncrono Trifásico)

Capítulo 9 TEMPORIZADORES

- 9.0 Generalidades
- 9.1 Imposição por Meio do Programa de um Determinado Tempo de Atraso
- 9.2 Timer de Pulso
- 9.3 Timer com Atraso no Desligamento
- 9.4 Timer com Atraso na Ligação com Memória
- 9.5 Timer com Atraso na Ligação
- 9.6 Operação de Temporização com a CPU S7-1200
 - 9.6.1 Timers TP, TON, TOF, TONR
 - 9.6.2 Base dos Tempos
 - 9.6.3 Exemplos de TP, TON, TOF, TONR
- 9.7 Energização de uma Saída com Timer TP
- 9.8 Aplicação: Lubrificação de um Motor Elétrico de Grande Potência
- 9.9 Aplicação: Circuito Eletropneumático Relativo a Pausa no Fim do Curso Fca1 de um Cilindro, Antes do Retorno Automático (Eletroválvula 5/2 Biestável e Cilindro de Duplo Efeito – DE)
- 9.10 Aplicação: Utilização do Timer TONR
- 9.11 Geração de Sinal de Duração Prestabelecida
- 9.12 Aplicação: Esteira Transportadora com Lampejo

Capítulo 10 CONTADORES

- 10.0 Generalidades
- 10.1 Contador Crescente/Decrescente
- 10.2 Ativação de um Contador
- 10.3 Operação de Contagem na CPU S7-1200
 - 10.3.1 Contadores CTU, CTD, CTUD
 - 10.3.2 O Campo Numérico de Contagem
 - 10.3.3 Exemplo de Uso do Contador CTUD
- 10.4 Aplicação: Enchimento de uma Caixa de Cerveja

Capítulo 11 APLICAÇÕES PRÁTICAS

- 11.0 Generalidades
- 11.1 Cablagem do PLC S7-1200
- 11.2 Consideração sobre as Entradas do PLC
- 11.3 Aplicação: Escada Rolante
- 11.4 Aplicação: Linha de Embalagem Automática

Bibliografia

Índice

Material Suplementar

Este livro conta com o seguinte material suplementar:

- Ilustrações da obra em formato de apresentação (restrito a docentes)

O acesso ao material suplementar é gratuito, bastando que o leitor se cadastre em: <http://gen-io.grupogen.com.br>.



GEN-IO (GEN | Informação Online) é o repositório de materiais suplementares e de serviços relacionados com livros publicados pelo GEN | Grupo Editorial Nacional, maior conglomerado brasileiro de editoras do ramo científico-técnico-profissional, composto por Guanabara Koogan, Santos, Roca, AC Farmacêutica, Forense, Método, LTC, E.P.U. e Forense Universitária. Os materiais suplementares ficam disponíveis para acesso durante a vigência das edições atuais dos livros a que eles correspondem.

LÓGICA PROGRAMADA

LÓGICA PROGRAMADA

- 1.0 Definição de *Programmable Logic Controller*
 - 1.1 Evolução dos Modernos PLC
 - 1.2 Vantagens e Precauções na Utilização do PLC
 - 1.3 Confiabilidade e Segurança no Sistema PLC
 - 1.4 Significado de Hardware e Software
 - 1.5 Descrição do Sistema PLC
 - 1.6 Considerações Finais
-

1.0 Definição de Programmable Logic Controller

O *Programmable Logic Controller* (PLC) é um equipamento composto de componentes eletrônicos, memória programável ou não programável que contém dados e programas com a finalidade de ler e executar instruções, interagindo com um sistema que deve ser controlado, através de dispositivos de input e output do tipo digital ou analógico (norma IEC 61131-1).

O PLC é um computador que realiza funções de controle em vários níveis de complexidade. O aspecto interessante do PLC é que ele pode ser programado e utilizado por pessoas sem um grande conhecimento no uso do computador. Esse pequeno computador (PLC) é projetado para trabalhar em ambiente industrial com variação de temperatura, umidade, vibrações, distúrbios elétricos e outras variantes que existem no ambiente industrial.

1.1 Evolução dos Modernos PLC

O primeiro PLC surgiu no final dos anos 1960 na indústria automotiva com a finalidade de abreviar os tempos de parada das máquinas na produção. Naqueles anos a indústria automotiva suspendia a atividade produtiva toda vez que mudava a produção de um modelo de montagem do mesmo tipo de automóvel. Ou seja, mudar o modelo de peça a ser produzida significava mudar todos os quadros elétricos, o que gerava um custo elevado para a empresa. A utilização dessa nova tecnologia permite que o processo de substituição do quadro elétrico não seja mais necessário, possa ser agora executado com uma simples programação do dispositivo PLC.

Os novos PLCs reduzem os tempos de parada da máquina de modo a tornar-se competitivos no mercado. Com o passar do tempo o

LÓGICA PROGRAMADA

- 1.0 Definição de *Programmable Logic Controller*
 - 1.1 Evolução dos Modernos PLC
 - 1.2 Vantagens e Precauções na Utilização do PLC
 - 1.3 Confiabilidade e Segurança no Sistema PLC
 - 1.4 Significado de Hardware e Software
 - 1.5 Descrição do Sistema PLC
 - 1.6 Considerações Finais
-

1.0 Definição de Programmable Logic Controller

O *Programmable Logic Controller* (PLC) é um equipamento composto de componentes eletrônicos, memória programável ou não programável que contém dados e programas com a finalidade de ler e executar instruções, interagindo com um sistema que deve ser controlado, através de dispositivos de input e output do tipo digital ou analógico (norma IEC 61131-1).

O PLC é um computador que realiza funções de controle em vários níveis de complexidade. O aspecto interessante do PLC é que ele pode ser programado e utilizado por pessoas sem um grande conhecimento no uso do computador. Esse pequeno computador (PLC) é projetado para trabalhar em ambiente industrial com variação de temperatura, umidade, vibrações, distúrbios elétricos e outras variantes que existem no ambiente industrial.

1.1 Evolução dos Modernos PLC

O primeiro PLC surgiu no final dos anos 1960 na indústria automotiva com a finalidade de abreviar os tempos de parada das máquinas na produção. Naqueles anos a indústria automotiva suspendia a atividade produtiva toda vez que mudava a produção de um modelo de montagem do mesmo tipo de automóvel. Ou seja, mudar o modelo de peça a ser produzida significava mudar todos os quadros elétricos, o que gerava um custo elevado para a empresa. A utilização dessa nova tecnologia permite que o processo de substituição do quadro elétrico não seja mais necessário, possa ser agora executado com uma simples programação do dispositivo PLC.

Os novos PLCs reduzem os tempos de parada da máquina de modo a tornar-se competitivos no mercado. Com o passar do tempo o PLC transformou-se em um equipamento mais simples de usar, mais confiável e produtivo. Dos anos 1980 em diante, ocorreu um enorme desenvolvimento na indústria dos computadores. O PLC tornou-se especializado, e cresceu exponencialmente sua difusão.

O mercado do PLC cresceu de um volume de cerca de 120 milhões de dólares em 1978 para cerca de 100 bilhões de dólares nos primeiros anos da década de 1990, e ainda hoje está crescendo.

1.2 Vantagens e Precauções na Utilização do PLC

Vantagens

- **Flexibilidade:** No passado a máquina automática requeria um próprio quadro elétrico feito de componentes eletromecânicos projetados exatamente para aquele tipo de máquina. Variar o funcionamento da máquina significava mudar todo o quadro elétrico de comando. Hoje, basta somente variar o programa (*software*), com um simples *personal computer* (PC), sem substituir o quadro elétrico, o que garante uma redução de custo considerável.
- **Fácil gestão das falhas:** Os modernos programas de PLC conseguem diagnosticar falhas de maneira rápida e simples em qualquer problema de automação industrial.
- **Grande número de contatos:** Um PLC tem um número de contatos teoricamente ilimitado, enquanto nos circuitos eletromecânicos tradicionais (o relé) o número de contatos é geralmente limitado.
- **Baixo custo:** Hoje é possível adquirir um pequeno PLC por algumas centenas de dólares.
- **Observação do funcionamento:** As operações de um PLC podem ser observadas por um monitor de computador durante sua execução. Pode ser testado previamente no PC mediante simulação e definitivamente quando se executa sua tarefa na instalação.
- **Velocidade da operação:** Na maioria das vezes o PLC é mais veloz em executar uma tarefa de automação industrial do que em um quadro de comando eletromecânico.
- **Facilidade de programação:** Um PLC pode ser programado por um técnico ou engenheiro que trabalha com mecânica, eletrônica e eletrotécnica.

Precauções

- **Nova tecnologia:** Nem sempre o trabalhador da indústria ou empresa quer converter seu modo de pensar e adequar-se a essa nova

tecnologia (fator humano).

- **Aplicação fixa:** Algumas aplicações têm uma única função que não pode ser mudada. Nesse caso não é conveniente utilizar um PLC. Exemplo: uma simples chave estrela/triângulo para um motor trifásico ou uma simples eletrobomba, assim como outros.
- **Condições ambientais extremas:** Lugares com alta temperatura ou com forte distúrbio eletromagnético que torna impossível utilizar um dispositivo eletrônico.

1.3 Confiabilidade e Segurança no Sistema PLC

A experiência desses últimos 20 anos tem demonstrado que o PLC tem um elevado grau de confiabilidade, naturalmente mais elevado que num sistema em lógica eletromecânica. E, como já foi dito anteriormente, o defeito do funcionamento da instalação é facilmente localizado e identificado graças ao software de controle e autodiagnóstico de que o PLC moderno dispõe.

1.4 Significado de Hardware e Software

Pela palavra *hardware* se entende tudo que se pode tocar fisicamente num PLC (fontes de alimentação, módulos de expansão, CPU e outros). Para permitir, porém, que o hardware funcione são necessários programas chamados *software de sistema*, escrito exclusivamente pelo construtor do PLC, geralmente não acessível a qualquer pessoa, e o *software usuário*, que é escrito por uma pessoa comum (usuário), o programador do PLC. O PLC, como qualquer outro computador, é uma máquina que sozinha não sabe executar nenhuma tarefa sem que antes seja inserido um programa que o transforme em uma máquina funcionante. Para comunicar esse programa ao PLC é necessário utilizar uma linguagem que a máquina entenda, chamada *linguagem de programação*.

1.5 Descrição do Sistema PLC

Como já relatado, os Controladores Lógicos Programáveis são indicados com a sigla PLC por causa das iniciais inglesas de **Programmable Logic Controller**. Foram idealizados e projetados para eliminar os inconvenientes característicos da lógica cabeada a relé. Desfrutando da potencialidade oferecida pela tecnologia eletrônica e em particular pelo uso de microprocessadores, os PLCs são capazes de garantir tarefa impensável para qualquer equipamento eletromecânico, dentre as quais se destacam:

- elaboração de sinais analógicos;
- realização de operações matemáticas;
- armazenamento de dados;
- visualização de dados;
- transferência de dados;
- conexão em rede do PLC com computadores, CNC (controle numérico) e outro equipamento.

Um PLC pode ser definido como um complexo sistema eletrônico para uso industrial capaz de gerir qualquer operação de controle de maneira flexível.

Como funciona? O PLC opera por meio da elaboração de sinais de entrada (sejam analógicos ou digitais) que chegam de sensores ou transdutores. Emite sinais de saída, sejam do tipo analógico ou digitais, com a modalidade preestabelecida por um programa. A emissão desses sinais de saída permite o comando dos atuadores. A Figura 1.1 representa a cabeagem de um PLC com dispositivos de entrada/saída e um pequeno teclado para programação (programador).

1.6 Considerações Finais

Este livro é inteiramente dedicado ao estudo do novo controlador lógico programável da série SIMATIC S7-1200 da multinacional Siemens, com o intuito de conduzir o leitor na realização de pequenas tarefas de automação industrial que utilizam esse equipamento. Vimos até agora que eles são sempre mais utilizados para realizar os *blocos de comando* de uma instalação automatizada acoplada ao *bloco de potência*, independentemente do tipo construtivo (elétrica, pneumática, hidráulica, mista).

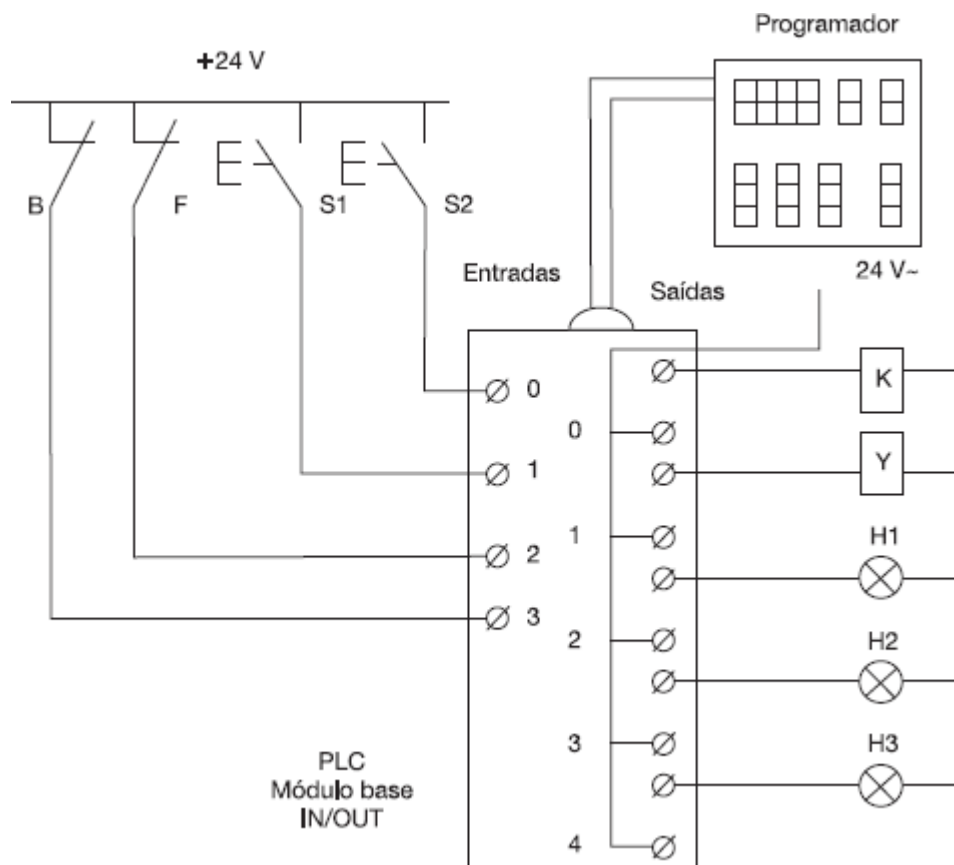


FIGURA 1.1

Na Figura 1.2 é mostrada uma comparação entre um comando eletromecânico e um comando eletrônico a lógica programável. O padrão atingido significa também confiabilidade e baixo custo, e esses são os motivos pelos quais o PLC representa, hoje, a solução ótima de muitos problemas de automação, em todos os níveis e em todos os setores industriais.

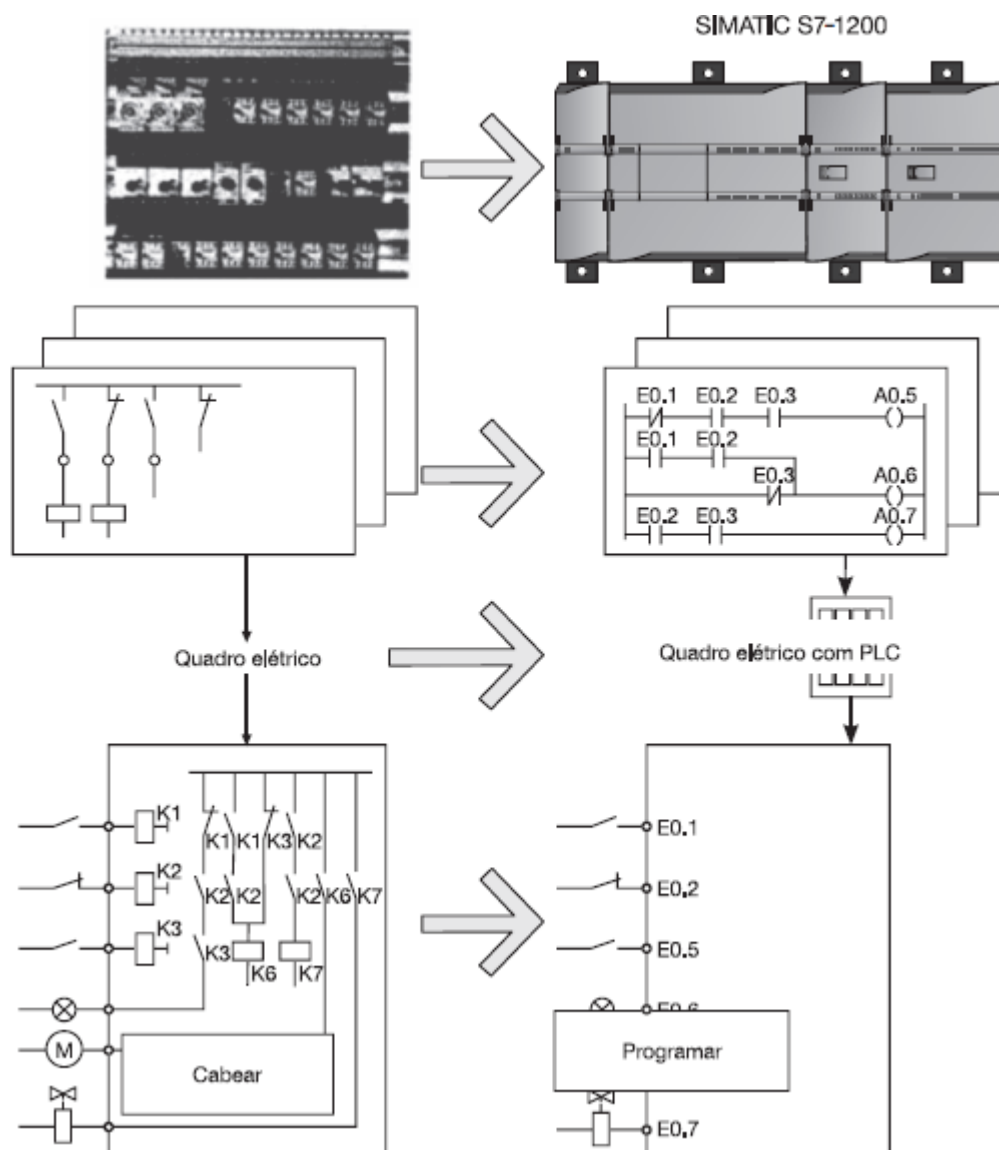


FIGURA 1.2 Comparação entre um comando eletromecânico e um comando eletrônico de lógica programável.

ARQUITETURA DO SISTEMA SIMATIC S7-1200

ARQUITETURA DO SISTEMA SIMATIC S7-1200

2.0 Generalidades

2.1 Os Componentes do Sistema S7-1200

2.0 Generalidades

O sistema de automação SIMATIC S7-1200 consiste em quatro CPUs do tipo: SIMATIC S7-1211C, S7-1212C, S7-1214C, S7-1215C e em painéis visualizadores da série SIMATIC HMI Basic Panels ligados entre si, com a rede chamada de SIMATIC NET. Todos os dispositivos são configurados e programados com uso de uma única plataforma software chamada de STEP 7 Basic.

Na Figura 2.1 temos os componentes básicos do sistema de automação SIMATIC S7-1200.

PLC SIMATIC S7-1200

São controladores lógicos programáveis que representam o “cérebro” do sistema de automação. Dispomos de quatro diferentes tipos de CPUs com diferentes tipos de desempenhos.

SIMATIC HMI

Deriva da sigla inglesa de “Human Machine Interface” e é constituído de painéis operadores para controle, visualização e monitoramento das variáveis no interior do sistema de automação.

A série SIMATIC HMI Basic Panels é projetada para uma perfeita integração com o PLC SIMATIC S7-1200.

Esses painéis operadores possuem displays de vários tamanhos operando com teclas do tipo touchscreen.

ARQUITETURA DO SISTEMA SIMATIC S7-1200

2.0 Generalidades

2.1 Os Componentes do Sistema S7-1200

2.0 Generalidades

O sistema de automação SIMATIC S7-1200 consiste em quatro CPUs do tipo: SIMATIC S7-1211C, S7-1212C, S7-1214C, S7-1215C e em painéis visualizadores da série SIMATIC HMI Basic Panels ligados entre si, com a rede chamada de SIMATIC NET. Todos os dispositivos são configurados e programados com uso de uma única plataforma software chamada de STEP 7 Basic.

Na Figura 2.1 temos os componentes básicos do sistema de automação SIMATIC S7-1200.

PLC SIMATIC S7-1200

São controladores lógicos programáveis que representam o “cérebro” do sistema de automação. Dispomos de quatro diferentes tipos de CPUs com diferentes tipos de desempenhos.

SIMATIC HMI

Deriva da sigla inglesa de “Human Machine Interface” e é constituído de painéis operadores para controle, visualização e monitoramento das variáveis no interior do sistema de automação.

A série SIMATIC HMI Basic Panels é projetada para uma perfeita integração com o PLC SIMATIC S7-1200.

Esses painéis operadores possuem displays de vários tamanhos operando com teclas do tipo touchscreen.

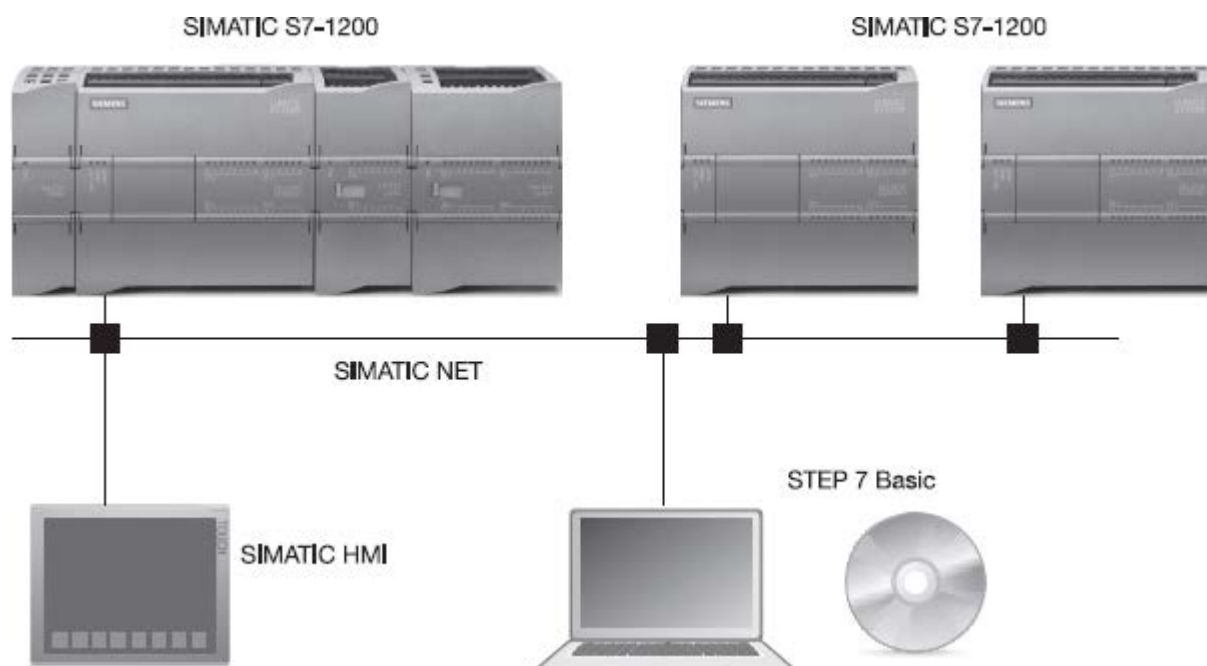


FIGURA 2.1

Protocolo de Comunicação

Permite a comunicação de todos os dispositivos do sistema S7-1200 com uso da interface PROFINET utilizada na rede de automação industrial padrão Ethernet para troca de dados e informações entre CPUs, painéis de operação HMI e outros dispositivos.

Com o uso de módulos de comunicação específicos, podemos utilizar o tipo de comunicação ponto a ponto com um só dispositivo utilizando os protocolos RS232, RS485, PROFIBUS e ASI.

Plataforma de software para configuração, programação e gerenciamento de todos os componentes do sistema SIMATIC S7-1200 e HMI.

Esse conceito é expresso com a sigla inglesa de *TIA portal* (sistema integrado completo de automação, do inglês *Totally Integrated Automation*).

2.1 Os Componentes do Sistema S7-1200

Os componentes básicos que compõem o sistema SIMATIC S7-1200 são descritos detalhadamente a seguir.

CPU

A **CPU** é o dispositivo mais complexo do sistema S7-1200. Esse dispositivo cumpre operações lógicas e matemáticas muito rapidamente e, sobretudo, as executa conforme uma sequência predefinida por nós, chamada simplesmente de programa. O elenco de todas as operações que um PLC pode executar é chamado set de instruções (do inglês *instruction set*).

O programa, portanto, não é outra coisa que um elenco de operações que o microprocessador deve cumprir.

Entre as muitas instruções que o microprocessador pode elaborar está também a da lógica binária (veja operação de tipo AND, OR). Ele, portanto, pode ser utilizado para elaborar sinais elétricos e comandar as máquinas, da mesma maneira como se faz em eletromecânica com o circuito elétrico. O suporte eletrônico que inclui o microprocessador é comumente chamado CPU (do inglês *Central Processing Unit*). A CPU é a parte mais importante do PLC; dela depende todo o desempenho do PLC, principalmente a sua velocidade e potência. Nesse momento podemos, portanto, definir o princípio de funcionamento do PLC.

Ao ler os sinais de entrada, a CPU carrega as informações provenientes da máquina obtidas por meio de um botão, interruptor de posição, sensores e/ou transdutor. Se existe uma variação de sinal (por exemplo, para acionamento de um botão), a CPU reage elaborando novos dados, segundo a lógica interna do programa, e gera os sinais na amplitude certa de saída. O sinal de saída comanda o dispositivo atuador, permitindo, assim, o movimento de uma válvula pneumática, um motor elétrico e outros conforme a sequência desejada.

Para esclarecer ainda mais o modo de funcionamento do PLC, consideremos o exemplo da Figura 2.2. Trata-se de um circuito muito simples que ilustra bem o funcionamento do PLC. Os dispositivos chamados de **campo** são nesse caso um interruptor I e uma lâmpada L, que devem acender quando I é fechado, e, ao contrário, devem-se apagar quando I é aberto.

Nesse caso I constitui a entrada do PLC e O, a saída.

O programa escrito instrui a CPU, informando-a não apenas que o interruptor I está fechado, mas que deve ligar a lâmpada L. A CPU executa o programa repetitivamente, e portanto, nesse caso, lê continuamente a entrada I até que I = 0 (interruptor aberto), assim L = 0 e não apenas I = 1 (interruptor fechado) habilita a saída L = 1, ligando assim a lâmpada.

O número de entradas e saídas disponíveis no módulo CPU, conforme a Figura 2.2, é geralmente limitado. Daqui para a frente utilizamos a expressão *On Board* para identificar somente as entradas e saídas disponíveis no módulo CPU.

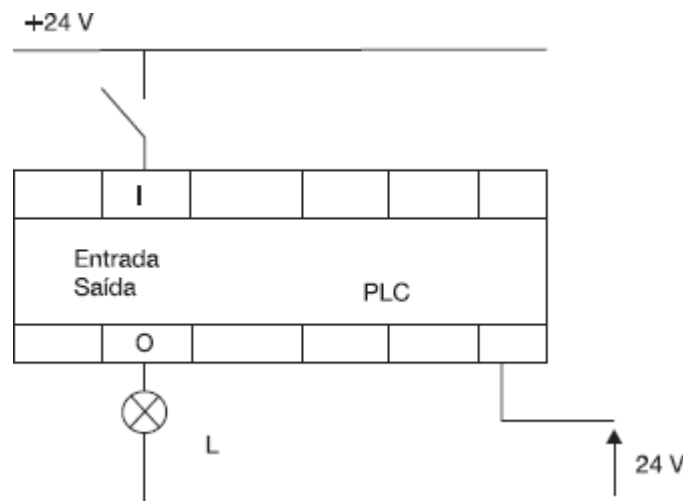


FIGURA 2.2

Memory card

Fornece a memória alternativa para o armazenamento do programa usuário. No caso de uso da memory card, a CPU executa o programa partindo da mesma memory card, e não da própria memória interna.

Signal Board (SB)

A Signal Board pode ser inserida diretamente na parte frontal da CPU. É possível assim acrescentar entradas e saídas suplementares do tipo analógicas e discretas I/O (Input/Output) sem aumentar o espaço físico disponível no módulo CPU.

Signal modules (SM)

Quando o número de entradas e saídas do tipo analógicas e discretas I/O (Input/Output) é maior que aquele disponibilizado pela CPU (On Board) ou da Signal Board, devemos utilizar os Signal modules (SM), ou seja, módulos adicionais de entradas e saídas analógicas e discretas.

PROFINET Interface

Porta de comunicação da CPU com outros dispositivos tais como: CPU, HMI, unidades remotas, acionamentos, etc. O protocolo de comunicação é o Ethernet industrial. Temos velocidade de transmissão dos dados na ordem de 10/100 Mbits por segundo.

No caso de precisarmos ligar a CPU a mais de um dispositivo por meio da rede PROFINET, o cabo deve ser roteado via multiplexador; em poucas palavras, precisamos de uma chave (switch) de alta velocidade tipo **switch module (CSM)**, exceto a CPU 1215C, que possui duas portas PROFINET integradas, podendo assim ligar até dois dispositivos.

Communication modules

Permitem a ligação ponto a ponto (*point to point*) dos dispositivos com a clássica porta serial RS232 ou RS485, para ligar dispositivos suplementares tipo código de barra etc.

A Figura 2.3 apresenta as conexões de todos os dispositivos citados anteriormente com a CPU SIMATIC S7-1200.

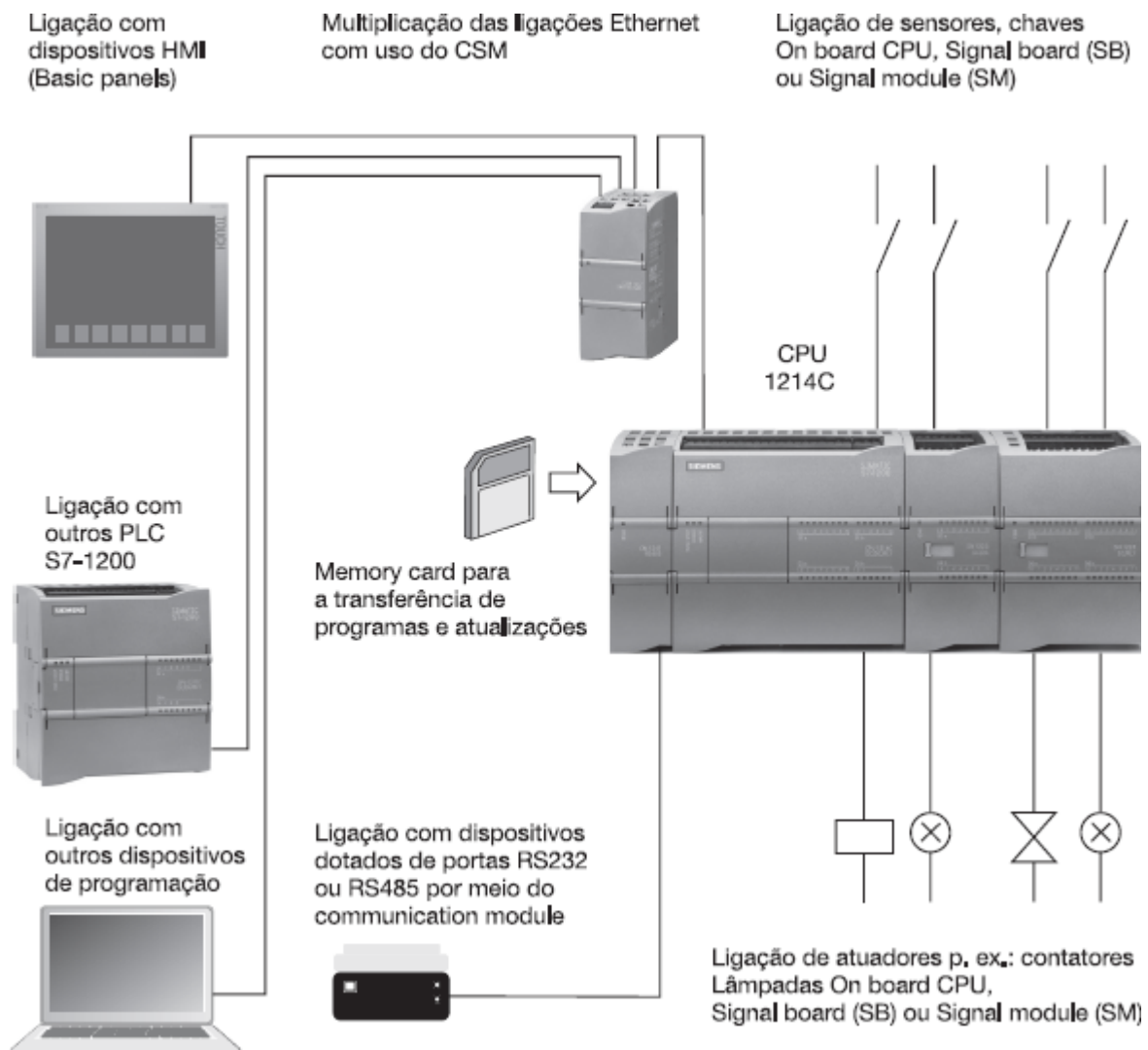


FIGURA 2.3

HARDWARE DO PLC S7-1200

HARDWARE DO PLC S7-1200

- 3.0 Generalidades
 - 3.1 Módulo CPU S7-1200
 - 3.2 Ligação com a Porta PROFINET
 - 3.3 Os Módulos de Comunicação (CM)
 - 3.4 Módulo Compact Switch (CSM)
 - 3.5 Módulo Power Supply (PM)
 - 3.6 Simulador de Entradas
 - 3.7 O Signal Module (SM)
 - 3.8 Características Técnicas Principais das CPUs S7-1200
 - 3.9 HMI Basic Panels
-

3.0 Generalidades

Na Figura 3.1 é apresentado o sistema completo de automação SIMATIC S7-1200.

HARDWARE DO PLC S7-1200

- 3.0 Generalidades
 - 3.1 Módulo CPU S7-1200
 - 3.2 Ligação com a Porta PROFINET
 - 3.3 Os Módulos de Comunicação (CM)
 - 3.4 Módulo Compact Switch (CSM)
 - 3.5 Módulo Power Supply (PM)
 - 3.6 Simulador de Entradas
 - 3.7 O Signal Module (SM)
 - 3.8 Características Técnicas Principais das CPUs S7-1200
 - 3.9 HMI Basic Panels
-

3.0 Generalidades

Na Figura 3.1 é apresentado o sistema completo de automação SIMATIC S7-1200.

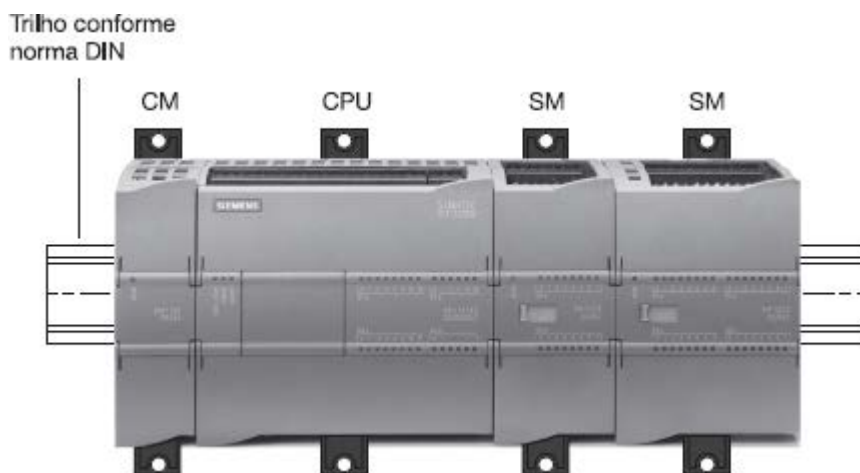


FIGURA 3.1

Conforme sugerido pela Siemens, à esquerda da CPU temos os módulos de comunicação (CM) e à direita os módulos de entrada e saída discretos ou analógicos (SM).

A instalação é simples e rápida; de fato, todo hardware do SIMATIC S7-1200 é dotado de engate rápido para trilho conforme norma DIN de 35 mm. A montagem pode ser horizontal ou vertical.

A instalação sem o trilho é possível devido ao fato de que os engates rápidos são retráteis.

Temos também outros acessórios externos, não indicados na Figura 3.1, tal como o módulo para fonte de alimentação externa, o switch multiplexador CSM para conexão PROFINET e o simulador de entrada. A seguir faremos uma apresentação de cada um desses componentes.

3.1 Módulo CPU S7-1200

O módulo CPU apresenta-se como na Figura 3.2. A seguir listamos suas partes fundamentais:

1. bornes de alimentação (Fase + Neutro);
2. bornes extraíveis de entrada e saída on board;
3. leds de estado das entradas e saídas;
4. porta PROFINET.

Como já relatado no capítulo anterior, temos quatro CPUs: SIMATIC S7-1211C, S7-1212C, S7-1214C, S7-1215C, utilizáveis em cada caso na versão tipo DC/DC/DC, AC/DC/RLY e DC/DC/RLY (o termo RLY indica a palavra relé). Veja a Figura 3.2.

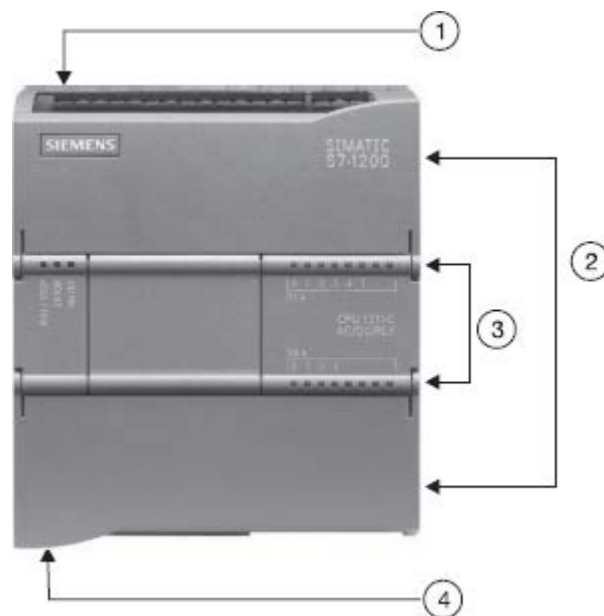


FIGURA 3.2

O primeiro termo à esquerda indica a fonte de alimentação da CPU (DC = 24 volts em corrente contínua, AC = 120/230 VAC volts em corrente alternada).

O termo no meio indica o tipo de alimentação das entradas digitais na CPU, ou seja, on board (DC = 24 volts em corrente contínua).

O termo à direita indica o tipo de alimentação das saídas digitais na CPU, ou seja, on board (DC = 24 volts em corrente contínua do tipo estático transistor, MOSFET – RLY = a relé, acima de 30 volts em corrente contínua ou máximo de 250 volts em corrente alternada).

As CPUs se diferenciam por tipo de fonte de alimentação, números de entradas e saídas on board, tamanho da memória e número máximo de módulos adicionais de entrada e saída.

O módulo CPU não possui a chave de on/off, e o modo operacional (RUN/STOP) deve ser selecionado obrigatoriamente pelo programa STEP 7 Basic.

3.1.1 Entradas e Saídas I/O — On Board

As entradas digitais DI (digital inputs) são alimentadas a 24 volts em corrente contínua 24 DC; cada CPU possui um número de entradas diferentes on board conforme o tipo de CPU utilizada.

O estado das entradas é visível por meio de LEDs.

As saídas digitais DO (digital outputs) trabalham com saídas estáticas a 24 volts em corrente contínua; 24 DC é corrente de 0,5 A com carga resistiva de 5 watts.

As saídas digitais DO do tipo relé trabalham acima de 30 VDC com corrente contínua de 2 A com carga resistiva de 30 watts ou no máximo de 250 VAC e corrente de 2 A com carga resistiva de 200 watts.

Cada CPU possui várias saídas diferentes on board conforme o tipo de CPU utilizada.

O estado das saídas é visível por meio de LEDs.

Cada CPU possui dois canais analógicos (input channel) on board com campo de medida de 0 a 10 volts. A resolução do conversor é geralmente de 10 bits. O valor numérico de conversão do sinal analógico varia de 0 a 27648.

Isso significa, em poucas palavras, que ao valor de medida de 0 volt o conversor quantifica o valor numérico zero, enquanto ao valor de medida de 10 volts o conversor quantifica o valor numérico de 27648.

Todas as entradas e saídas são dotadas de parafusos extraíveis. Portanto, é suficiente cabear de uma só vez todo o equipamento, com consequente economia de tempo na fase de instalação.

3.1.2 O Signal Board (SB)

Como já abordado no capítulo anterior, o signal board (SB) expande no próprio módulo CPU as entradas e saídas suplementares do tipo analógicas e discretas I/O (input/output), sem aumentar o espaço físico disponível no módulo CPU. O slot a ser inserido é localizado na frente do módulo CPU, conforme apresentado na Figura 3.3.

O SB usa as entradas e saídas de 24 VDC e 5 VDC, e algumas dessas entradas e saídas operam com uma frequência até 200 kHz, ou seja, podem ser utilizadas pela contagem veloz ou como saídas impulsivas.

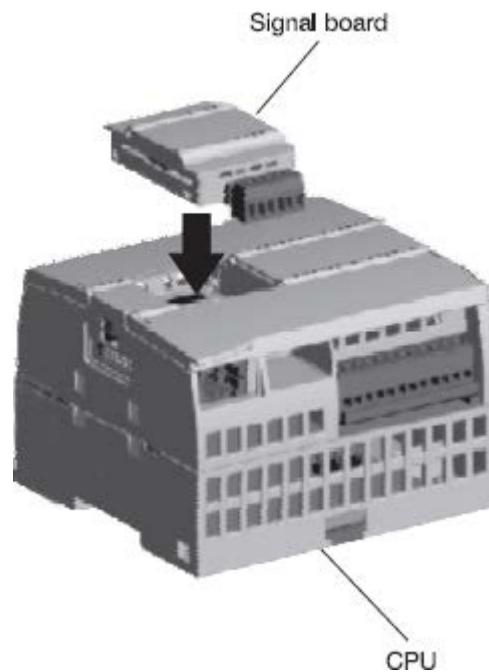


FIGURA 3.3

Temos também a possibilidade de ter uma saída analógica de 12 bits no campo de medida de ± 10 volts, ou 0 – 20 mA.

3.1.3 Modo Operacional da CPU

O estado da CPU é indicado com três LEDs na frente do módulo (veja Figura 3.4):

- **Led RUN/STOP:** a cor amarela corresponde à condição de STOP, o verde, à condição de RUN, e o lampejo, à condição de encaminhamento (START UP).
- **Led ERROR:** lampejo vermelho indica uma condição de falha no sistema.
- **Led MAINT:** a cor amarela indica modo de funcionamento estável.

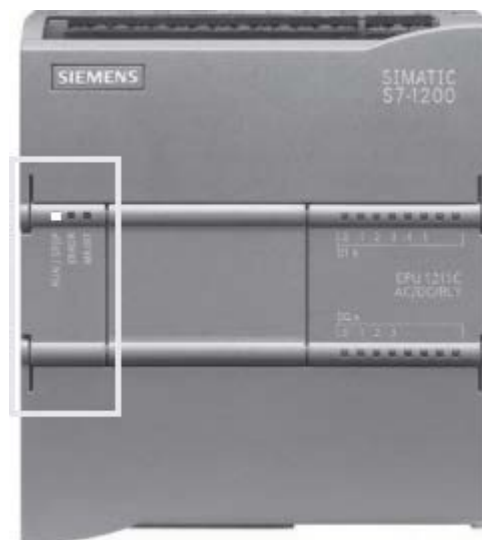


FIGURA 3.4

Sobre o módulo CPU não está presente fisicamente a chave de comutação RUN/STOP, então para comutar devemos utilizar o programa STEP 7 Basic.

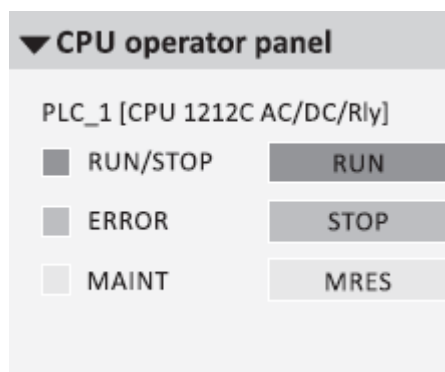


FIGURA 3.5

A janelinha “CPU operator panel” (veja a Figura 3.5) permite a comutação do PLC na condição de:

- **STOP:** nesse caso a CPU não executa nenhum programa, com a possibilidade de monitorar e executar o download do programa usuário.
- **RUN:** o ciclo de scan da CPU é executado continuamente. Os eventos de alarme podem ser executados a qualquer momento.
- **MRES:** reset completo da memória.

3.1.4 Ciclo de Scan

Vimos no capítulo anterior que a CPU faz todas as suas operações de acordo com uma ordem sequencial à qual se dá o nome **ciclo de scan** ou **varredura**, esquematizado na Figura 3.6.

A cada leitura das entradas, os dados são armazenados em um registro particular da memória, chamado **imagem do processo das entradas**. O mesmo acontece com os dados de saída: antes de serem colocados nas saídas externas, eles são armazenados em um registro particular da memória chamado **imagem do processo das saídas**. (Lembramos que por registro se entende um conjunto de células de memória.)

Todas as operações apresentadas na Figura 3.6 constituem um **ciclo de scan**, e o tempo para executá-lo é chamado **tempo de scan**.

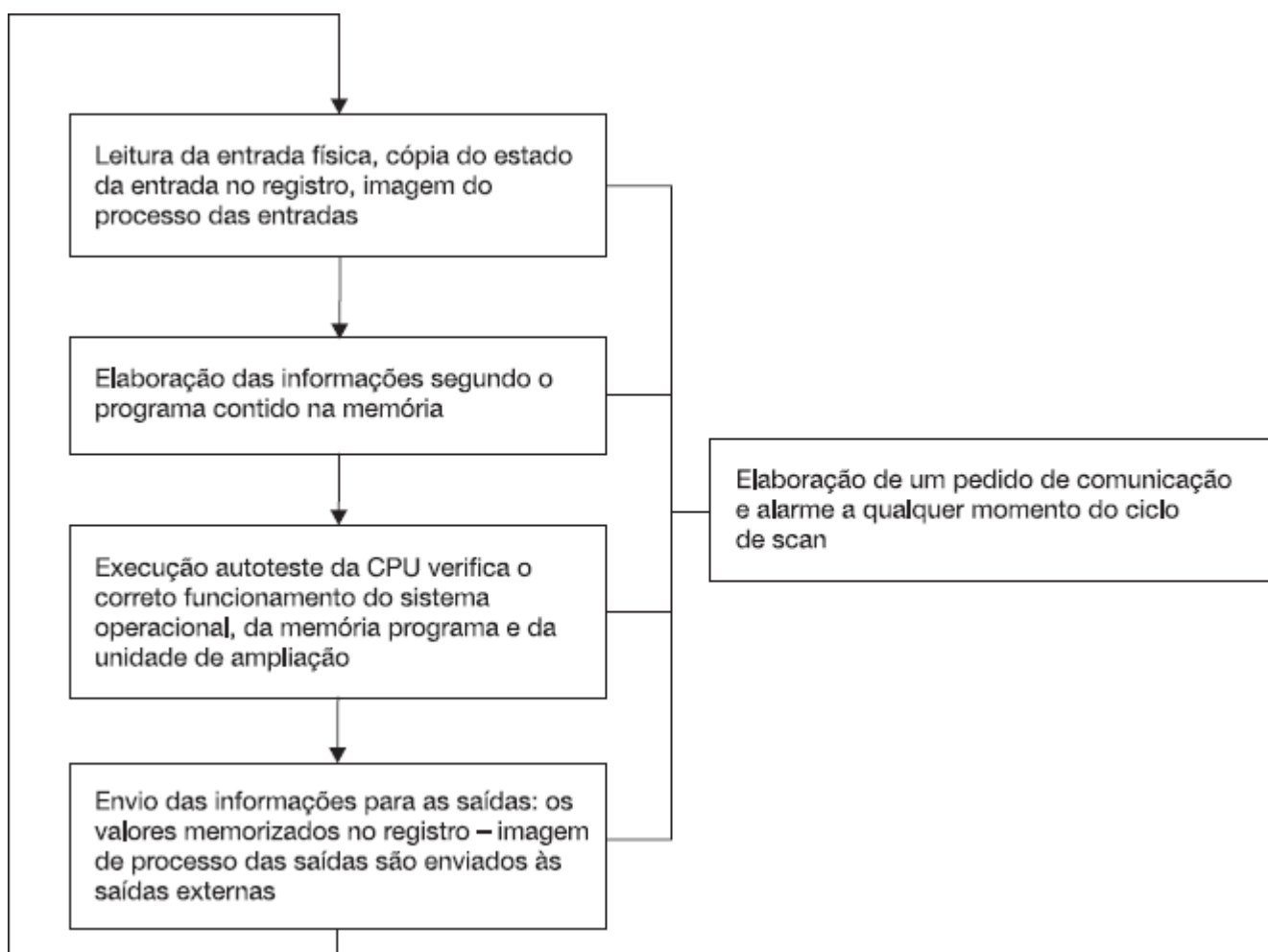


FIGURA 3.6

3.1.5 A SIMATIC Memory Card

Fornece a memória alternativa para o armazenamento do programa usuário, ou para atualizações do próprio sistema operacional

(firmware).

No caso de uso do memory card, a CPU carrega o programa para execução do memory card, em vez da própria memória interna (veja a Figura 3.7).



FIGURA 3.7

Temos duas versões da SIMATIC Memory Card: com capacidade de 2 Mbytes ou 24 Mbytes.

3.2 Ligação com a Porta PROFINET

É possível o uso da porta PROFINET integrada ao sistema SIMATIC S7-1200, para a programação e a comunicação com dispositivos HMI e demais controladores.

Por meio de PROFINET, é possível a conexão de dispositivos de campo (chão de fábrica) com o sistema SIMATIC S7-1200.

3.2.1 Interface Integrada PROFINET

O novo controlador SIMATIC S7-1200 dispõe de uma ou duas portas integradas PROFINET, que ligam a comunicação com os demais dispositivos (veja a Figura 3.8).

Suporta a programação e a comunicação com os SIMATIC HMI Basic Panels e a possibilidade de ligação entre CPU e CPU e demais controladores, com possíveis integrações ainda maiores.



FIGURA 3.8

A interface de comunicação é constituída de uma conexão do tipo RJ45 com elevada imunidade aos distúrbios e funcionalidade do tipo autocrossover. Suporta até 16 ligações PROFINET, com uma velocidade de transmissão de dados de até 10/100 Mbytes por segundo.

Para minimizar a cabeagem e permitir a máxima flexibilidade da rede, podemos empregar o novo módulo CSM 1277 Compact Switch, que, ligado ao controlador SIMATIC S7-1200, permite ligar outros dispositivos com a máxima simplicidade.

3.3 Os Módulos de Comunicação (CM)

Cada CPU SIMATIC S7-1200 pode ser ampliada com módulos de comunicação (CM, do inglês communication modules), a serem inseridos na parte à esquerda do módulo CPU (veja a Figura 3.9).

Os módulos de comunicação do tipo RS232 e RS485 são para a ligação serial ponto a ponto ASCII.

A comunicação é programada e configurada com simples instruções utilizando a biblioteca USS drive protocol e Modbus RTU Master e slave protocol, contida no programa STEP 7 Basic.

O módulo de comunicação CM 1241 possui as características relacionadas a seguir:

- **CM 1241 RS232:** ligação ponto a ponto e com protocolo Modbus.
- **CM 1241 RS485:** ligação ponto a ponto e com protocolo Modbus, USS drive protocol.



FIGURA 3.9

Com os controladores S7-1200 firmware na versão 2.0 e suas sucessivas versões, foram acrescentados dois módulos de comunicação para a ligação na rede de automação PROFIBUS e nas áreas remotas com os módulos de comunicação GRPS/GSM, providenciando a ligação entre telemóveis e redes (networks).

– Módulo PROFIBUS CM 1243-5, CM 1242-5

Para a comunicação com PROFIBUS, temos o módulo CM 1243-5, PROFIBUS

DP master (mestre), que permite a ligação com até 16 DP slaves (escravos) a um PLC S7-1200, a módulos distribuídos I/O (SIMATIC ET 200) e comunicação PG/OP com outros controladores, bem como os dispositivos HMI. Com o módulo CM 1242-5 o SIMATIC S7-1200 pode funcionar como PROFIBUS DP slave, podendo assim ser ligado a qualquer controlador PROFIBUS master.

– Módulo AS-i CM 1243-2

Lembramos que AS-i é um protocolo de comunicação aberto e independente do fabricante de hardware, para a ligação de atuadores e sensores no sistema automático conforme a norma europeia EN-50295.

O módulo de comunicação CM 1243-2 realiza como mestre AS-i uma interface entre o controlador S7-1200 e sensores/atuadores, por exemplo, módulos I/O, chaves de posição, indicadores luminosos, partidas de motores etc.

O módulo CM 1243-2 precisa ser implementado de forma correta, em conjunto com o módulo de desacoplamento de dados DCM 1271, permitindo assim uma perfeita integração com o controlador S7-1200. Com um único módulo CM 1243-2 podemos controlar até 62 escravos AS-i ou 992 I/O, transmissão integrada do valor analógico, parametrização de soft starter motores AS-i SIRIUS M200D (correntes motores, valores limites etc.) sem módulos funcionais suplementares no controlador. A manutenção e o diagnóstico de mestre AS-i e escravo AS-i podem ser efetuados com uma simples configuração do STEP 7 Basic no TIA Portal.

– Módulo CM 1242-7 GSM

Os novos módulos CM 1242-7, Teleservice Adapter IE Basic e Teleservice Module GSM permitem aos operadores o acesso remoto aos controladores SIMATIC S7-1200. Dessa forma, com o SIMATIC S7-1200 é possível monitorar, diagnosticar e controlar, de forma rentável, estações remotas. O CM 1242-7 é ligado através do bus interno do SIMATIC S7-1200 como interface GSM/GPRS e usado para telecontrole. O Teleservice Adapter IE Basic e o Teleservice Module GSM estão disponíveis para manutenção e diagnósticos remotos. Para utilizar os novos serviços de telecomunicações, os utilizadores acessam através de um contrato convencional utilizando um cartão SIM, ou através de um contrato especial M2M (máquina a máquina). Assim, têm acesso remoto ao controlador utilizando um PC equipado com o STEP 7 Basic V11 e acesso à Internet.

3.4 Módulo Compact Switch (CSM)

Para ligar no máximo três dispositivos ao PLC SIMATIC S7-1200 com a rede PROFINET podemos empregar o módulo CSM 1277 Compact Switch; as configurações possíveis são do tipo linear, a estrela ou árvore (veja a Figura 3.10).

A ligação se atua com uso de quatro portas RJ45; nelas podemos ligar dispositivos HMI, controladores programáveis e outros dispositivos com uma velocidade de transmissão de dados de até 10/100 Mbytes por segundo.

3.5 Módulo Power Supply (PM)

Lembramos brevemente que o PLC S7-1200 é ligado diretamente à tensão de rede e pode alimentar também outro aparelho que faz parte do sistema que está sendo controlado. Ele providencia automaticamente as tensões e a corrente para o circuito interno da CPU e para os módulos de expansão, garantindo ainda o isolamento elétrico. Se a corrente absorvida pelos sensores envolvidos no processo, ou eventuais módulos de I/O, ultrapassam um determinado limite, deve ser utilizada uma fonte de alimentação externa.



FIGURA 3.10

A fonte de alimentação externa power supply PM 1207 possui uma alimentação de 120/230 VAC, frequência 50/60 Hz e saída de 24 VDC.

A corrente máxima de saída é de 2,5 ampères com 24 VDC (veja a Figura 3.11).



FIGURA 3.11

3.6 Simulador de Entradas

O simulador de entradas é um dispositivo conectável ao controlador para efetuar testes do programa.

Existem várias soluções construtivas. A Figura 3.12 mostra um modelo diretamente aplicável aos miniparafusos das entradas do PLC.

O simulador é muito simples: trata-se de um bloco, no qual é montado certo número de chaves que simulam o funcionamento do sensor.

O simulador é utilizado porque em fase de projeção e simulação não se dispõe de todos os sensores necessários. Isso ocorre com muita frequência, sobretudo, em laboratórios didáticos. Basicamente, o simulador pode ser extraído do miniparafuso do PLC com uma simples chave de fenda pequena.

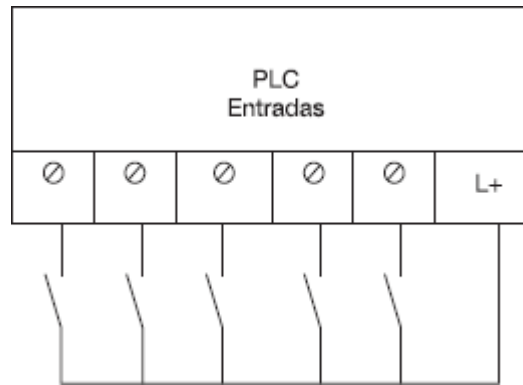


FIGURA 3.12

A Figura 3.13 mostra o simulador de entradas ligado ao PLC S7-1200.

O simulador da CPU 1212C possui oito chaves, conforme a Figura 3.13, enquanto na CPU 1214C e CPU 1215C temos 14.

3.7 O Signal Module (SM)

Quando o número das entradas e saídas do tipo analógicas e discretas I/O (input/output) é maior que aquele disponibilizado pela CPU (on board) ou a Signal Board, devemos utilizar os signal modules (SM), ou seja, módulos adicionais de entradas e saídas analógicas e discretas.

Os signal modules são inseridos à direita da CPU, e conforme o tipo de CPU podemos ligar:

- Na CPU 1211C, nenhum signal module.
- 1212C no máximo de dois signal modules.
- Na CPU 1214C, no máximo oito signal modules.

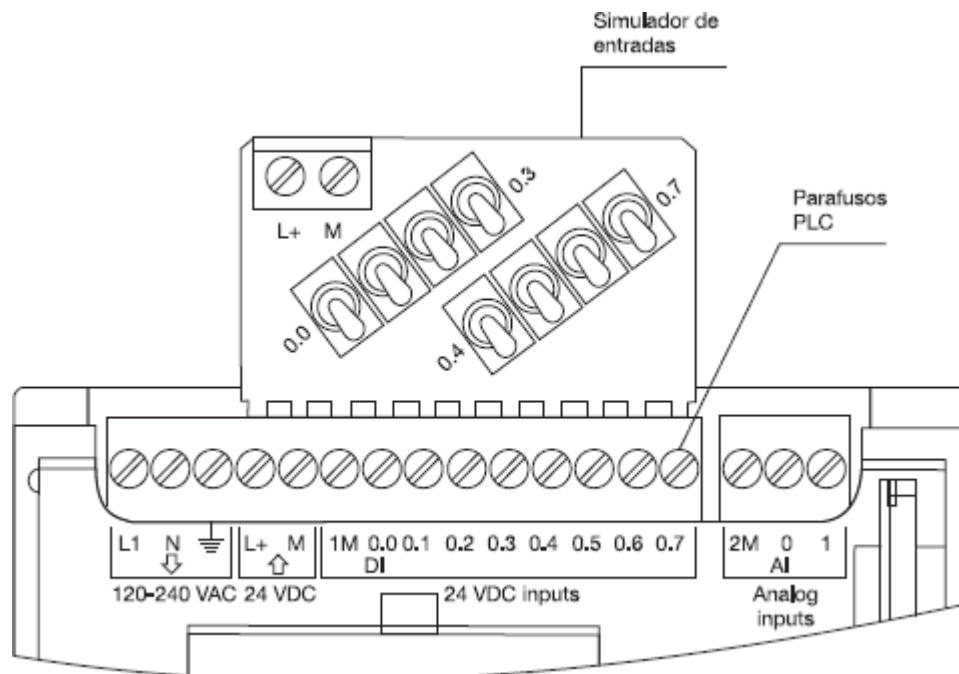


FIGURA 3.13

- Na CPU 1215C, no máximo oito signal modules.
- **Signal module digital:** O signal module digital de entrada converte um sinal externo de 24 VDC em um sinal compatível com os circuitos interiores do PLC.

Ao contrário, o signal module digital de saída converte um sinal interno do PLC em um sinal de 24 VDC do tipo estático (transistor - MOSFET) ou a relé.

O relé comuta uma carga em corrente contínua de no máximo 30 W, enquanto em corrente alternada é mais elevada, cerca de 200 W. Veja a Figura 3.14.

Todos os signal modules digitais são disponibilizados com 8 e 16 pontos de entrada.

– **Signal module analógico:** o signal module analógico serve para o processamento de sinais analógicos.

Por isso é provido de conversor A/D (analógico/digital) ou D/A (digital/analógico) para converter o sinal analógico no equivalente numérico (digital) para ser processado no interior do PLC. Veja a Figura 3.15.

O sinal analógico é muito utilizado nos processos industriais quando é necessário trabalhar com grandezas físicas, tais como temperatura, pressão, velocidade, aceleração e outros.

Cada ponto de entrada ou saída analógico é normalmente chamado de “canal”. Cada canal processa o valor analógico no equivalente numérico a 16 bits do tipo fixed-point (ponto fixo).



FIGURA 3.14



FIGURA 3.15

Os signal modules analógicos encontram-se disponíveis com dois, quatro e oito canais de entrada e saída.

3.7.1 Propriedade das Entradas e Saídas Digitais

Entrada Digital

Algumas entradas digitais do módulo CPU (on board) e do signal board (SB) podem ser usadas pela contagem veloz (HSC). A frequência é de cerca de 100 kHz; no caso de entradas digitais do módulo CPU e de cerca de 200 kHz no caso de entradas digitais do signal board(SB).

Saída Digital

Algumas saídas digitais do módulo CPU (on board) podem funcionar como trem de pulsos de alta frequência - PTO (pulse train output) ou como saída de pulsos modulada em largura PWM (pulse width modulation).

A sequência dos pulsos configurada como PTO oferece uma relação duty/cycle de 50% até 100 kHz para a regulação contínua da velocidade e posição com motores de passos ou acionamento com inversores de frequência.

É possível assim regular a velocidade de um motor, a posição de uma válvula hidráulica ou a temperatura de um forno.

A frequência é de cerca de 100 kHz no caso de saídas digitais do módulo CPU e de cerca de 200 kHz no caso de saídas digitais do signal board (SB).

3.7.2 Propriedade das Entradas e Saídas Analógicas

Entrada Analógica

Cada canal analógico de entrada pode ser configurado pelo programa STEP 7 Basic com vários alcances de medidas: em tensão de ± 10 V, ± 5 V, $\pm 2,5$ V e em corrente de 0 a 20 mA. O conversor possui uma resolução geralmente de 12 bits + sinal, com valor numérico entre -27648 e +27648.

O padrão 4 a 20 mA pode ser processado usando entradas de 0 a 20 mA.

Em poucas palavras, isso significa que a conversão chamada de bipolar (negativa ou positiva) do tipo em tensão de, por exemplo, ± 5 V; com a medida de -5 volts o conversor quantifica o valor numérico -27648, com +5 volts quantifica o valor numérico de +27648.

Saída Analógica

Cada canal analógico de saída possui dois alcances de medidas: em tensão de ± 10 V e em corrente de 0 a 20 mA.

No caso da medida em tensão de ± 10 V a resolução do conversor é de 14 bits, com valor numérico entre -27648 e +27648. A impedância como carga máxima admitida é maior ou igual a 1000 ohms.

No caso da medida em corrente de 0 a 20 mA, a resolução do conversor é de 13 bits, com valor numérico entre 0 e +27648. Nesse caso a impedância como carga máxima admitida é menor ou igual a 600 ohms.

3.7.3 O Novo Módulo SM 1231 4 × AI 16 Bits

Recentemente foi lançado um novo módulo analógico, SM 1231 4 × AI 16 bits, ou seja, um módulo analógico dotado de quatro entradas analógicas de alta resolução (16 bits), enquanto os anteriores possuem uma resolução máxima de 14 bits.

Os campos de medidas dos sinais analógicos são: em tensão de ± 10 V, ± 5 V, $\pm 2,5$ V, $\pm 1,25$ V e em corrente de 0 a 20 mA e 4 a 20 mA. O novo módulo possui o controle de quebra de fio.

3.8 Características Técnicas Principais das CPUs S7-1200

Apresentamos as características técnicas principais das CPUs S7-1200; para demais informações, a consulta ao manual de sistema é absolutamente indispensável.

Resumo das principais características da CPU S7-1211C

- Uma porta PROFINET integrada
- Memória: 25 KB de memória de trabalho, 1 MB de memória de carregamento, 2 KB de memória de dados retentivos
- Tempo de execução das operações booleanas: 0,1 μ S/instrução
- Duas entradas analógicas on board
- Seis entradas discretas on board/quatro saídas discretas on board
- Três contadores rápidos de 100 kHz
- Duas saídas de pulso PTO integradas
- Relógio integrado com backup de duração típica de 10 dias

Resumo das principais características da CPU S7-1212C

- Uma porta PROFINET integrada
- Memória: 25 KB de memória de trabalho, 1 MB de memória de carregamento, 2 KB de memória de dados retentivos
- Tempo de execução das operações booleanas: 0,1 μ S/instrução
- Duas entradas analógicas on board
- Oito entradas discretas on board/seis saídas discretas on board
- Quatro contadores rápidos (três contadores rápidos de 100 kHz, um de 30 kHz)
- Duas saídas de pulso PTO integradas
- Relógio integrado com backup de duração típica de 10 dias

Resumo das principais características da CPU S7-1214C

- Uma porta PROFINET integrada
- Memória: 50 KB de memória de trabalho, 2 MB de memória de carregamento, 2 KB de memória de dados retentivos
- Tempo de execução das operações booleanas: 0,1 μ S/instrução
- Duas entradas analógicas on board
- Quatorze entradas discretas on board/10 saídas discretas on board

- Seis contadores rápidos (três contadores rápidos de 100 kHz, três de 30 kHz)
- Duas saídas de pulso PTO integradas
- Relógio integrado com backup de duração típica de 10 dias

Resumo das principais características da CPU S7-1215C

- Duas portas PROFINET integradas
- Memória: 100 KB de memória de trabalho, 4 MB de memória de carregamento, 10 KB de memória de dados retentivos
- Tempo de execução das operações booleanas: 85 nS/instrução
- Duas entradas analógicas on board/duas saídas analógicas on board (ambas de 0-10 V)
- Quatorze entradas discretas on board/10 saídas discretas on board
- Seis contadores rápidos (três contadores rápidos de 100 kHz, três de 30 kHz)
- Quatro saídas de pulso PTO integrados para o controle simultâneo de quatro motores de passo
- Relógio integrado com backup de duração típica de 20 dias
- Atualização do firmware via porta PROFINET

É importante lembrar que as saídas de pulso PTO integradas são suportadas somente para controladores S7-1200 do tipo DC/DC/DC.

3.9 HMI Basic Panels

O sistema S7-1200 dispõe de uma série de painéis operadores para as principais operações de comando e controle por parte de um operador (veja a Figura 3.16).

Todos os painéis possuem uma série de teclas de cristal líquido, com um grau de proteção IP 65 e dotados de certificações CE, UL, cULus e NEMA 4x.

Todos os painéis operadores para o sistema S7-1200 são configurados por meio da mesma plataforma STEP 7 Basic, sem custo adicional no software de programação.

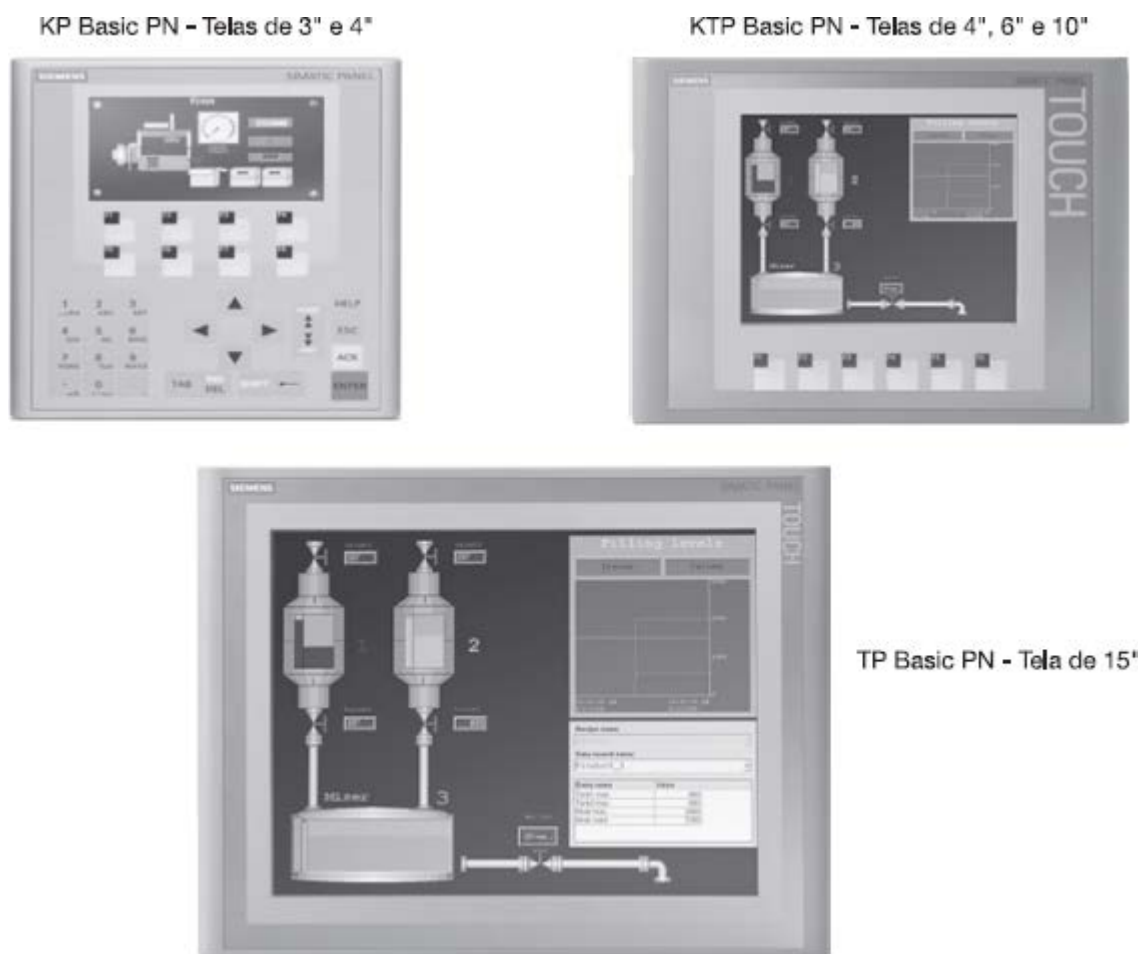


FIGURA 3.16

INTRODUÇÃO À LINGUAGEM DE PROGRAMAÇÃO CONFORME NORMA IEC 61131-3

INTRODUÇÃO À LINGUAGEM DE PROGRAMAÇÃO CONFORME NORMA IEC 61131-3

- 4.0 Generalidades
 - 4.1 A Norma IEC 61131-3
 - 4.2 Linguagem de Programação Textual
 - 4.3 Linguagem de Programação Gráfica
 - 4.4 Linguagem em Lista de Instruções – IL
 - 4.5 Linguagem com Texto Estruturado – ST
 - 4.6 Linguagem LADDER – LD
 - 4.7 Linguagem com Diagrama de Blocos Funcionais – FBD
-

4.0 Generalidades

Este capítulo apresenta as várias modalidades de programação do PLC conforme a norma internacional IEC 61131-3, destacando, no entanto, a mais usada, que demonstra na forma gráfica os esquemas elétricos ou diagramas com portas lógicas.

4.1 A Norma IEC 61131-3

A norma de referência para a programação dos PLCs é a IEC 61131-3, que resume todas as modalidades de programação dos PLCs. De todas as linguagens de programação, pode-se fazer uma primeira macrodistinção em duas grandes categorias:

INTRODUÇÃO À LINGUAGEM DE PROGRAMAÇÃO CONFORME NORMA IEC 61131-3

4.0 Generalidades

4.1 A Norma IEC 61131-3

4.2 Linguagem de Programação Textual

4.3 Linguagem de Programação Gráfica

4.4 Linguagem em Lista de Instruções – IL

4.5 Linguagem com Texto Estruturado – ST

4.6 Linguagem LADDER – LD

4.7 Linguagem com Diagrama de Blocos Funcionais – FBD

4.0 Generalidades

Este capítulo apresenta as várias modalidades de programação do PLC conforme a norma internacional IEC 61131-3, destacando, no entanto, a mais usada, que demonstra na forma gráfica os esquemas elétricos ou diagramas com portas lógicas.

4.1 A Norma IEC 61131-3

A norma de referência para a programação dos PLCs é a IEC 61131-3, que resume todas as modalidades de programação dos PLCs. De todas as linguagens de programação, pode-se fazer uma primeira macrodistinção em duas grandes categorias:

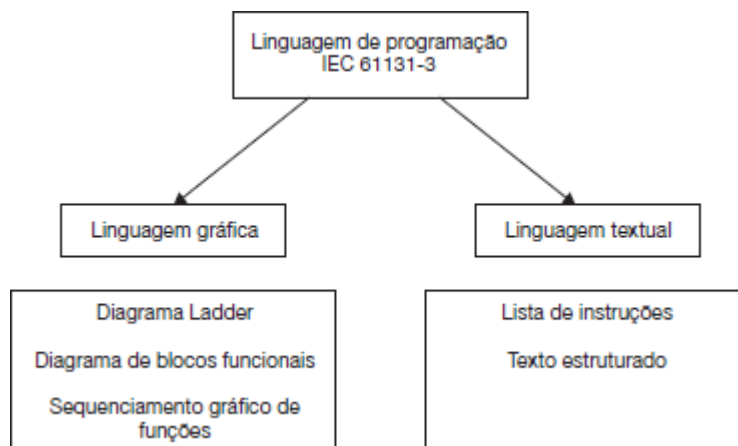


FIGURA 4.1

4.2 Linguagem de Programação Textual

Essa modalidade de programação é das mais “difíceis” para quem se propõe ao mundo do PLC proveniente do setor eletrotécnico. Tal sistema de programação provém do setor eletrônico ou então informático e, portanto, constitui a verdadeira linguagem de programação, com enorme possibilidade de desenvolver as necessidades do projeto.

– Lista de Instruções – IL (*Instruction List*)

Esse modo de programação é praticamente a linguagem de máquina muito parecida com a linguagem Assembler, ou seja, uma linguagem que usa diretamente as instruções do microcomputador. Tal sistema é pouco intuitivo e pouco prático, não tem nenhuma representação gráfica e requer muito tempo do programador para a pesquisa de falhas no programa.

Alguns PLCs não permitem o uso desse sistema; porém, com outros, ele é indispensável para se usar plenamente a potencialidade do PLC.

– Texto Estruturado – ST (*Structured Text*)

Essa linguagem de programação é de alto nível, como as linguagens Pascal, Basic e C++, e às vezes é indispensável em determinada aplicação ou nas redes de comunicação.

4.3 Linguagem de Programação Gráfica

A linguagem de programação gráfica se apresenta ao programador como um verdadeiro esquema elétrico ou esquema de blocos.

– Diagrama Ladder – LD (*Ladder Diagram*)

Esta é a linguagem de programação mais utilizada; assemelha-se a um esquema elétrico funcional. *Ladder* significa literalmente “escada”, dado que visualmente o esquema lembra uma escada. Cada degrau é chamado *rung*.

– Diagrama de Blocos Funcionais – FBD (*Function Block Diagram*)

É uma linguagem a “portas lógicas” que permite desenhar um esquema clássico de eletrônica digital. É muito utilizado nos sistemas de controle de grande instalação de processos (centrais termoelétricas, instalação química e outros).

– Sequenciamento Gráfico de Funções – SFC (*Sequential Function Chart*)

É uma linguagem desenvolvida na França com o nome de linguagem *Grafcet*. Representa o funcionamento por passos de um processo automático de modo muito similar aos diagramas de blocos já estudados. A única diferença é que cada bloco representa um estado do processo de trabalho da máquina automática.

4.4 Linguagem em Lista de Instruções – IL

Com já foi dito no parágrafo anterior, esta é uma linguagem do tipo textual. É muito potente e não utiliza símbolos gráficos. O aspecto negativo é que, ao contrário do que acontece com a linguagem gráfica, não se tem a visão rápida do funcionamento do programa. A linguagem IL contém linhas de instruções alfanuméricas. A sintaxe desse tipo de instrução até agora depende do tipo de PLC em que se está trabalhando. Os fabricantes estão, gradativamente, se alinhando ao padrão internacional IEC 61131-3.

O esquema a seguir demonstra um exemplo de programa escrito na linguagem IL, no qual são evidenciadas as características principais.

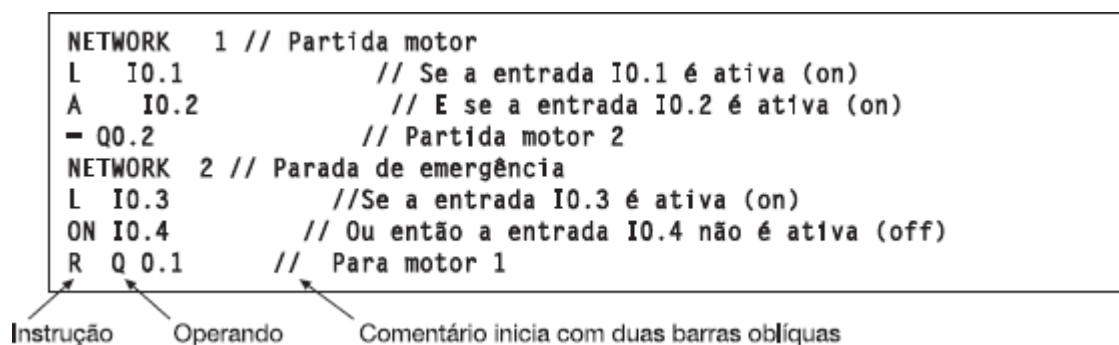


FIGURA 4.2

É bom precisar que a linguagem IL geralmente é a única compreensível diretamente do controlador (PLC), porque contém os códigos da linguagem “máquina”. No caso da linguagem gráfica, é o software da unidade de programação que a converte em linguagem “máquina”.

4.5 Linguagem com Texto Estruturado – ST

A linguagem com texto estruturado é aquela que tem uma base informática. Ela é, na verdade, a evolução da linguagem Pascal. Nessa linguagem aparecem estruturas do tipo IF, THEN, DO, ELSE e outras, típicas da programação de linguagem de alto nível. É sem dúvida a linguagem de programação mais potente para o PLC, porque oferece possibilidades que com outra linguagem seriam inexistíveis.

Como frisamos anteriormente, para desfrutar plenamente a potencialidade de tal linguagem é indispensável a utilização do diagrama de bloco já descrito. O uso dessa linguagem de programação pressupõe um conhecimento de informática muito aprofundado, mais familiar ao programador informático do que ao técnico elétrico ou mecânico. Na Figura 4.3 vemos um exemplo simples de programação com esse tipo de linguagem.

Diagrama de blocos

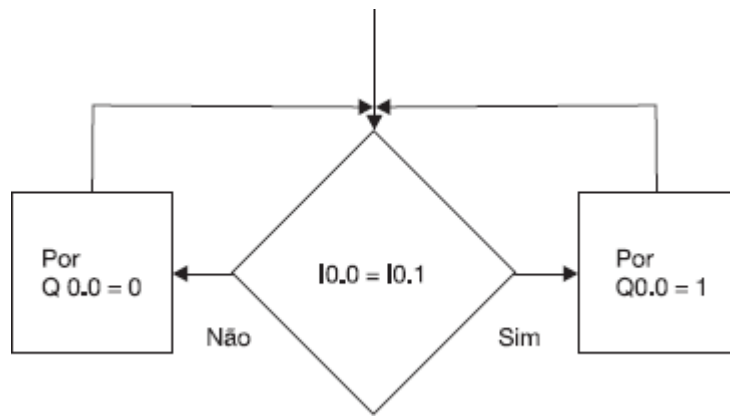


FIGURA 4.3

Tradução em texto estruturado

```

10 IF I0.0=I0.1    // Se a entrada I0.0 é igual à entrada I0.1
20 THEN SET Q0.0 // Ativa (on) a saída Q0.0
30 ELSE RES Q0.0 // Se não desativa (off) a saída Q0.0
40 GOTO 10        // Salta a instrução 10

```

4.6 Linguagem LADDER – LD

O nome deriva do inglês *ladder* (escada em degraus) pelo formato particular da representação. É usualmente denominada “esquema de contatos”. De todas as linguagens conhecidas hoje, é a mais empregada no campo internacional para a programação do PLC.

É composta de uma série de sinais gráficos e resulta preferida pelo programador com conhecimento da lógica eletromecânica porque requer um conhecimento do esquema a relé.

Esses sinais gráficos são padronizados pela norma IEC 61131-3 e derivam de uma simbologia americana já em uso para os esquemas funcionais.

Os esquemas Ladder são compostos estruturalmente de duas linhas verticais e de linhas horizontais (escada), sob as quais são desenhados os elementos constituintes do sistema a controlar. Veja a Figura 4.4.

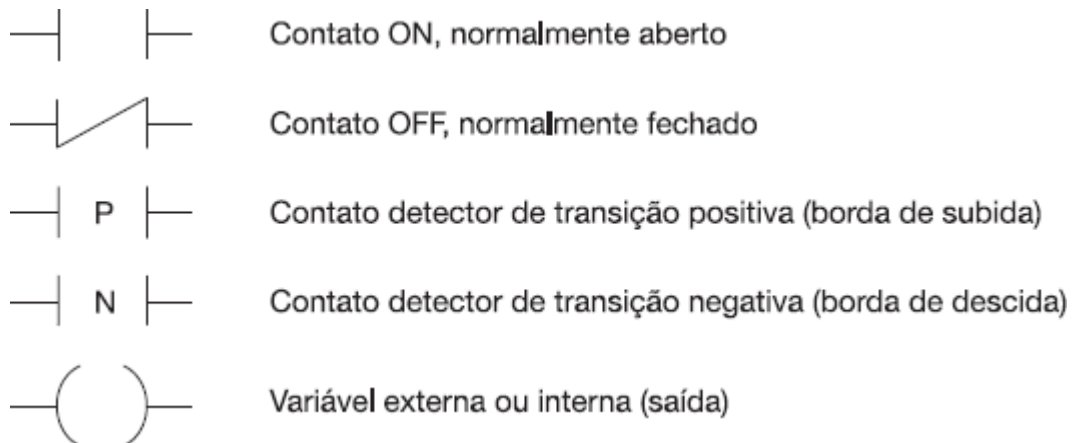


FIGURA 4.4

O esquema Ladder é composto:

- da barra de alimentação (linha vertical esquerda), que comanda todos os elementos de input;
- do retorno comum ou massa (linha vertical direita que frequentemente é omitida), que conecta todas as variáveis de saída (output);
- da zona de teste (ou zona de input). Nessa parte do esquema são desenhados os vários inputs em série e em paralelo em fechamento e abertura, segundo a lógica de evolução do processo;
- da zona de ação (a zona de output), destinada à variável de saída. Veja a Figura 4.5.

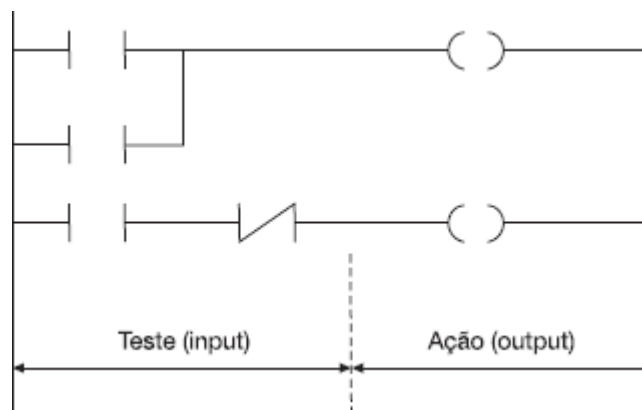


FIGURA 4.5

A Figura 4.6 apresenta um exemplo simples de programação.

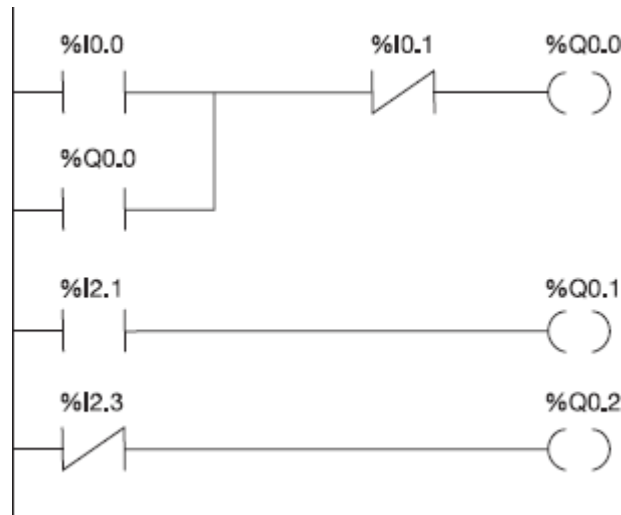


FIGURA 4.6

4.7 Linguagem com Diagrama de Blocos Funcionais – FBD

É uma linguagem composta de uma série de símbolos gráficos clássicos da lógica combinatória (AND, OR, XOR e outros). É mais usada pelos técnicos com experiência de eletrônica digital. Na Figura 4.7 são apresentados os símbolos gráficos básicos desse tipo de linguagem.

Na Figura 4.8 apresenta-se um circuito que tem como base de partida um esquema elétrico de circuito a relé com memória, com linguagem de blocos lógicos e Ladder.

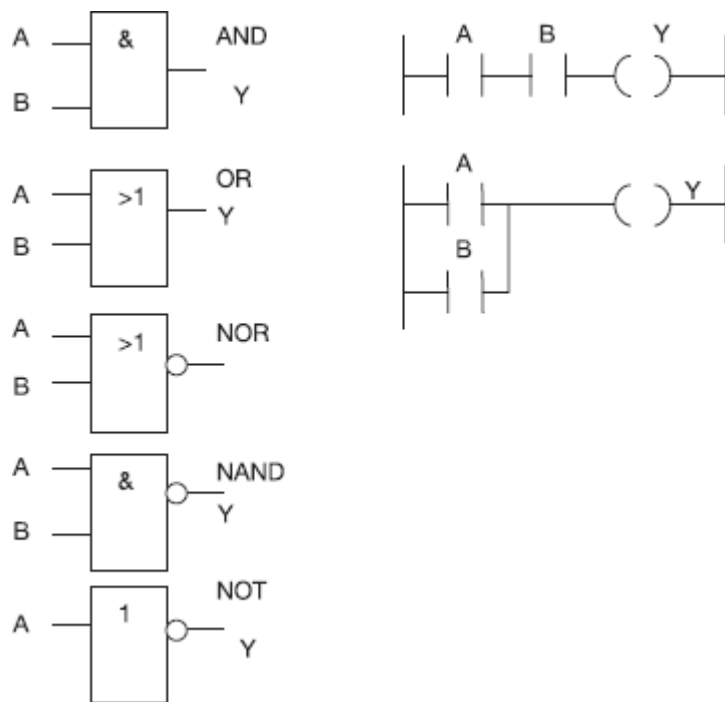


FIGURA 4.7

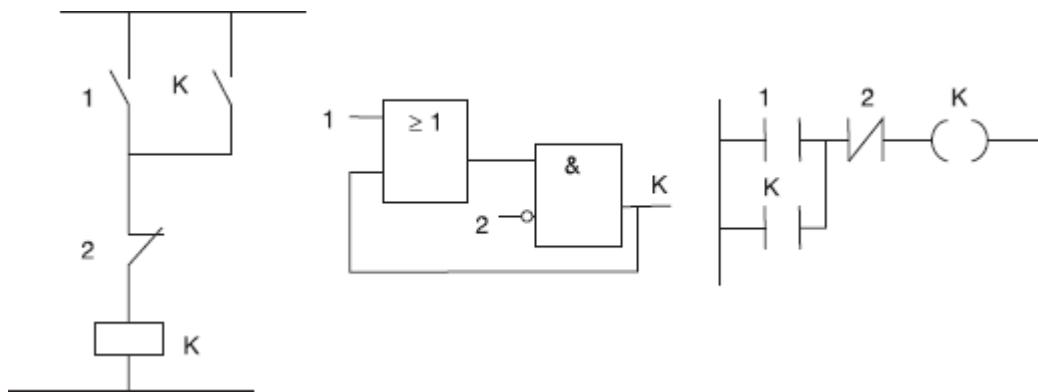


FIGURA 4.8

A Figura 4.9 é outro exemplo de confronto da linguagem FBD e linguagem Ladder.

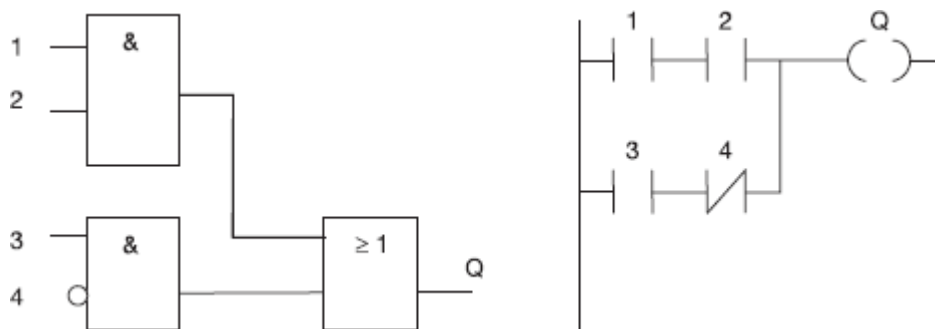


FIGURA 4.9

INTRODUÇÃO AO STEP 7 BASIC

INTRODUÇÃO AO STEP 7 BASIC

- 5.0 Generalidades
 - 5.1 As Novidades da STEP 7 V11.0
 - 5.2 Portal View (Visual Portal) e Project View de STEP 7 Basic
 - 5.3 Instalação de STEP 7 Basic
 - 5.4 Primeiros Passos no Portal View
 - 5.5 Guia On-line
 - 5.6 Editor de Programação
 - 5.7 Editor Tabela dos Símbolos
 - 5.8 Criação de um Projeto com STEP 7 Basic
 - 5.9 Exemplo de Criação de um Projeto
-

5.0 Generalidades

O software STEP 7 Basic V11.0 disponibiliza um ambiente de fácil utilização para o desenvolvimento da lógica de controle dos controladores S7-1200, configuração e visualização dos dispositivos HMI e definição da comunicação em rede.

O STEP 7 Basic oferece dois diferentes visuais do projeto que permitem trabalhar de forma mais eficiente:

- **Portal view** (Visual portal): orientado em função da tarefa a ser executada e das ferramentas.
- **Project view** (Visual do projeto): orientado mais ao projeto a ser executado, o usuário pode escolher entre as duas modalidades.

5.1 As Novidades da STEP 7 V11.0

A plataforma STEP 7 V11.0 está disponível com dois níveis de licença de uso:

1. STEP 7 Basic para a programação do PLC S7-1200
2. STEP 7 Professional para a programação de PLC S7-1200, S7-300, S7-400 & WinAC.

INTRODUÇÃO AO STEP 7 BASIC

5.0 Generalidades

- 5.1 As Novidades da STEP 7 V11.0
 - 5.2 Portal View (Visual Portal) e Project View de STEP 7 Basic
 - 5.3 Instalação de STEP 7 Basic
 - 5.4 Primeiros Passos no Portal View
 - 5.5 Guia On-line
 - 5.6 Editor de Programação
 - 5.7 Editor Tabela dos Símbolos
 - 5.8 Criação de um Projeto com STEP 7 Basic
 - 5.9 Exemplo de Criação de um Projeto
-

5.0 Generalidades

O software STEP 7 Basic V11.0 disponibiliza um ambiente de fácil utilização para o desenvolvimento da lógica de controle dos controladores S7-1200, configuração e visualização dos dispositivos HMI e definição da comunicação em rede.

O STEP 7 Basic oferece dois diferentes visuais do projeto que permitem trabalhar de forma mais eficiente:

- **Portal view** (Visual portal): orientado em função da tarefa a ser executada e das ferramentas.
- **Project view** (Visual do projeto): orientado mais ao projeto a ser executado, o usuário pode escolher entre as duas modalidades.

5.1 As Novidades da STEP 7 V11.0

A plataforma STEP 7 V11.0 está disponível com dois níveis de licença de uso:

1. STEP 7 Basic para a programação do PLC S7-1200
2. STEP 7 Professional para a programação de PLC S7-1200, S7-300, S7-400 & WinAC.

Todas as versões de STEP 7 V11.0 permitem o uso de:

- WinCCBasic, a ferramenta de programação e parametrização dos painéis operadores KTP Basic;
- Teleservice V11.0.

5.2 Portal View (Visual Portal) e Project View de STEP 7 Basic

O *Portal view* (Visual portal) disponibiliza as representações funcionais das tarefas (*tasks*) do projeto e organiza as funções das ferramentas na base da tarefa a ser executada (veja a Figura 5.1).

Saber como proceder e quais tasks escolher é muito fácil:

1. Portal para as diferentes tarefas;
2. Tarefa do portal selecionado;
3. Painel da ação selecionada;
4. Passagem ao Project view.

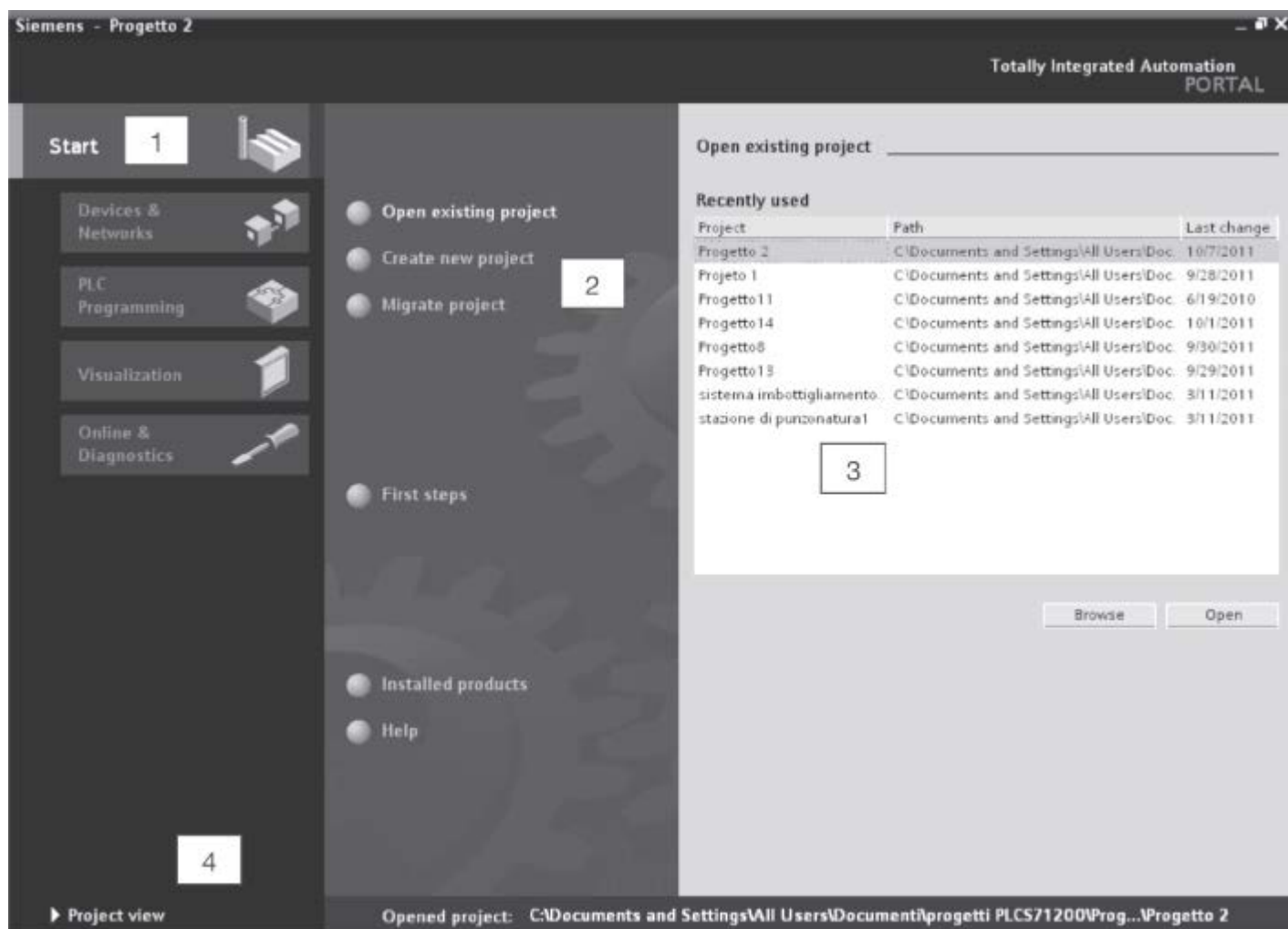


FIGURA 5.1

O Project view (Visual do projeto) permite o acesso aos componentes de um projeto (veja a Figura 5.2):

1. Barra das ferramentas;
2. Navegação de um projeto;
3. Área de trabalho (configuração hardware);
4. Catálogo hardware;
5. Janela de inspeção (visual geral dos dispositivos);
6. Informação;
7. Passagem ao Portal view.

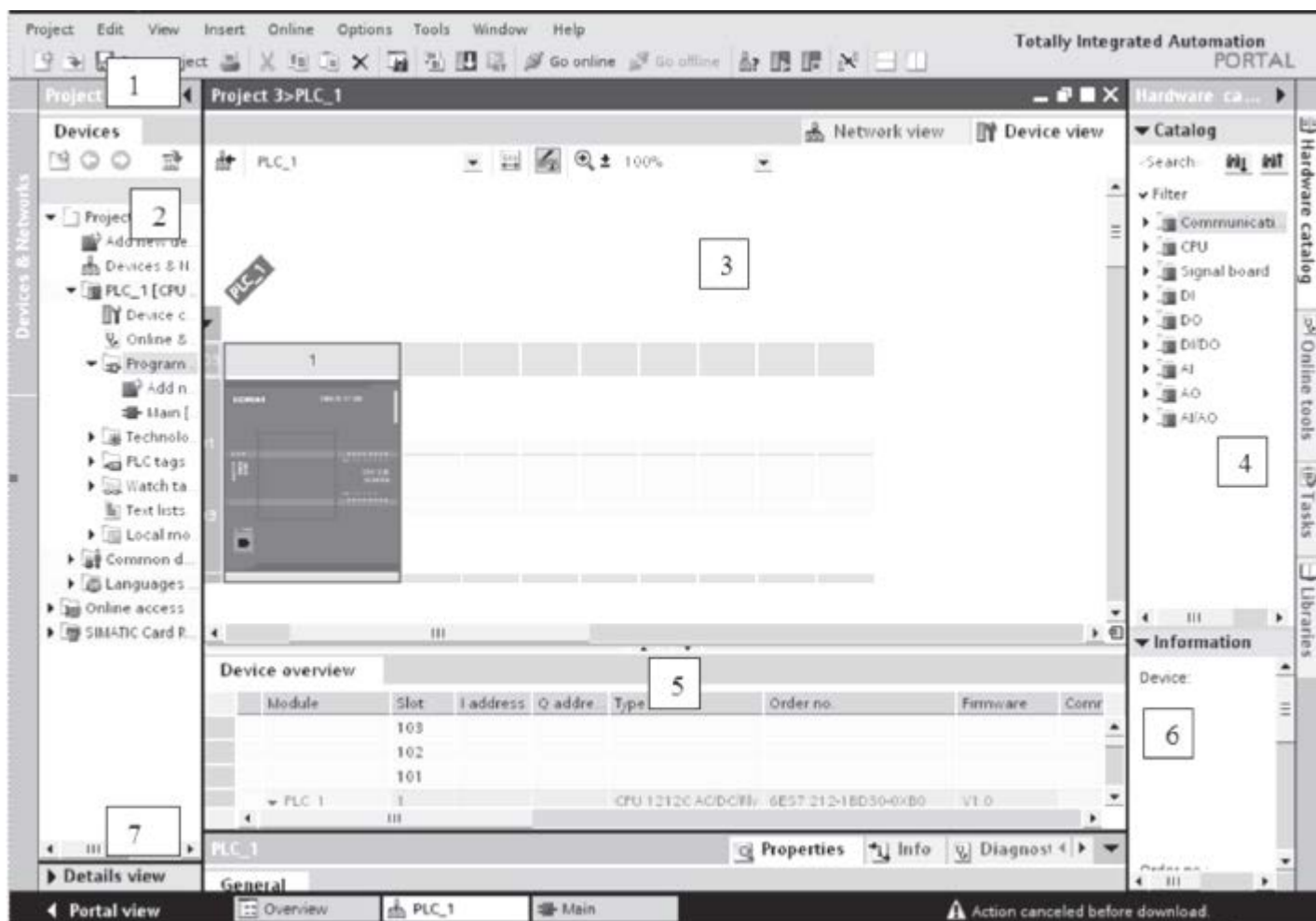


FIGURA 5.2

5.3 Instalação de STEP 7 Basic

O software STEP 7 Basic V11.0 é um aplicativo de 32/64 bits executado pelo sistema operacional Windows XP (Home ou Professional com SP3). É possível utilizar também o sistema operacional Windows 7. O processador aconselhado deve ser pelo menos do tipo Pentium 4 de 1,7 GHz, com 1 Gbyte de RAM.

Para poder instalar o novo STEP 7 V11.0 Professional ou Basic é necessário ter pelo menos os seguintes sistemas operacionais:

- Microsoft Windows XP Home SP3 (somente STEP 7 Basic);
- Microsoft Windows XP Professional SP3;
- Microsoft Windows 7 Home Premium (somente STEP 7 Basic);
- Microsoft Windows 7 Professional.

É importante lembrar que o STEP 7 Basic é acompanhado de um pen drive adicional com a licença de uso (License Manager). Tal licença pode ser transferida automaticamente no hard disk do computador no momento da instalação do STEP 7 Basic, ou pode ser transferida manualmente a qualquer momento com o software Automation License Manager. Após transferida a licença no pen drive, este não precisa mais estar conectado no computador.

Os requisitos de hardware para a nova versão STEP 7 V11.0 Professional ou Basic são os seguintes:

- Configuração mínima: processador Pentium 4 de 1,7 GHz ou equivalente, RAM 1 Gbyte, monitor 1024 × 768.
- Configuração aconselhada: processador Core Duo de 2 GHz ou equivalente, RAM 2 Gbytes, monitor 1280 × 1024.

5.4 Primeiros Passos no Portal View

Após a instalação do STEP 7 Basic V11.0 aparece a janela conforme Figura 5.3 do Portal view, permitindo a seleção das seguintes tarefas:

- Devices & Networks (Dispositivos e redes): o portal configura todos os componentes do hardware e do controlador programável e a própria ligação na rede de automação. Podemos então selecionar todos os componentes hardware do nosso sistema de automação, por exemplo, tipo de CPU, tipo e posição dos módulos adicionais no trilho DIN e propriedade de cada componente.

- PLC Programming (Programação do PLC): o seguinte portal permite a programação completa do controlador programável.
- Visualization (Visualização): permite a configuração completa dos Basic panels HMI, por exemplo, imagem visual do processo a ser controlado, mensagens etc.
- Online & Diagnostics (Online e diagnóstico): permite ligação do controlador programável, transferência e debug do programa, além de detectar falhas no sistema de automação.

Além disso, com a nova CPU S7-1200, podemos criar um novo projeto (Create new project), abrir um projeto já existente (Open existing project), fazer a migração de um projeto (Migrate project), ou seja, a conversão de projetos atuados com CPU anteriores (tipo CPU S7-200). Na janela à direita de Primeiros passos (First steps), com um clique do mouse entramos no editor Configura dispositivo (Configure device), Cria programa no PLC (Create a PLC program), Configura dispositivo HMI (Configure an HMI screen).

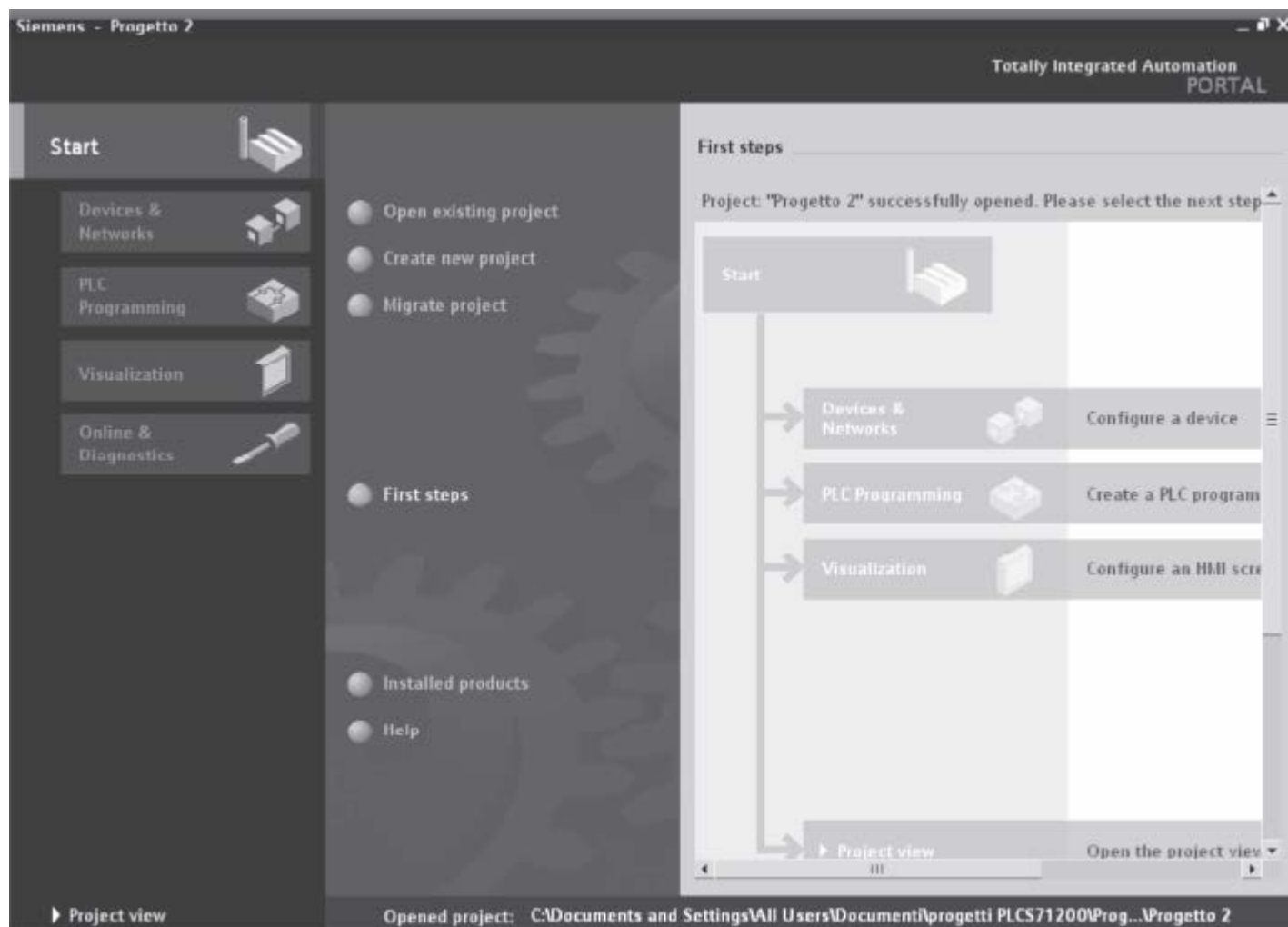


FIGURA 5.3

5.5 Guia On-line

Para informação e resolução de eventuais problemas de forma rápida e eficiente, é preciso ir à barra de ferramentas e clicar com o mouse no símbolo “?”; logo aparece a janela que pode ser visualizada na Figura 5.4. Temos assim um minucioso guia on-line.

5.6 Editor de Programação

Na plataforma STEP 7 Basic V11.0, estão disponíveis duas linguagens de programação, conforme norma IEC 61131-3:

1. **LAD** - Editor gráfico da linguagem Ladder (veja capítulo anterior).
2. **FBD** - Editor gráfico da linguagem com diagramas de blocos funcionais FBD (veja capítulo anterior).

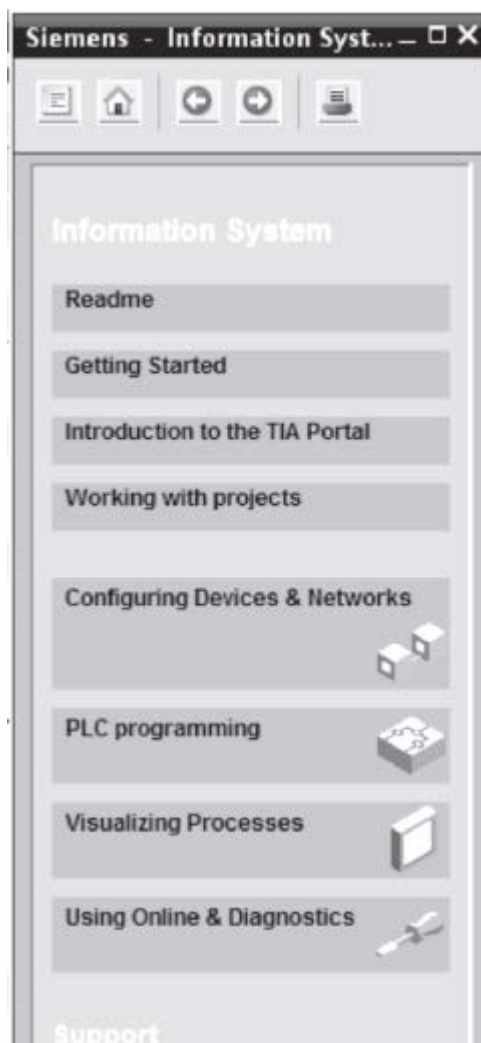


FIGURA 5.4

O STEP 7 Basic V11.0 converte automaticamente qualquer programa escrito em Ladder no equivalente FBD, e vice-versa.

5.7 Editor Tabela dos Símbolos

A Tabela dos símbolos permite ao programador utilizar o endereçamento simbólico. No PLC existem o *endereçamento absoluto* e o *endereçamento simbólico*. O endereçamento absoluto é o endereço que o PLC reconhece como código de máquina. Exemplo: %I0.0 é uma entrada absoluta. Na maioria das vezes, usar %I0.0 para uma entrada dificulta a identificação, mas se usarmos símbolos tipo S1, que pode significar, por exemplo, botão de parada, torna-se mais simples identificar tal símbolo. No ícone Tabela dos símbolos é possível identificar S1 no lugar de %I0.0, ou seja, utiliza-se o endereçamento simbólico no lugar do absoluto. O programa compilado e carregado na CPU converte automaticamente o endereço simbólico em endereço absoluto.

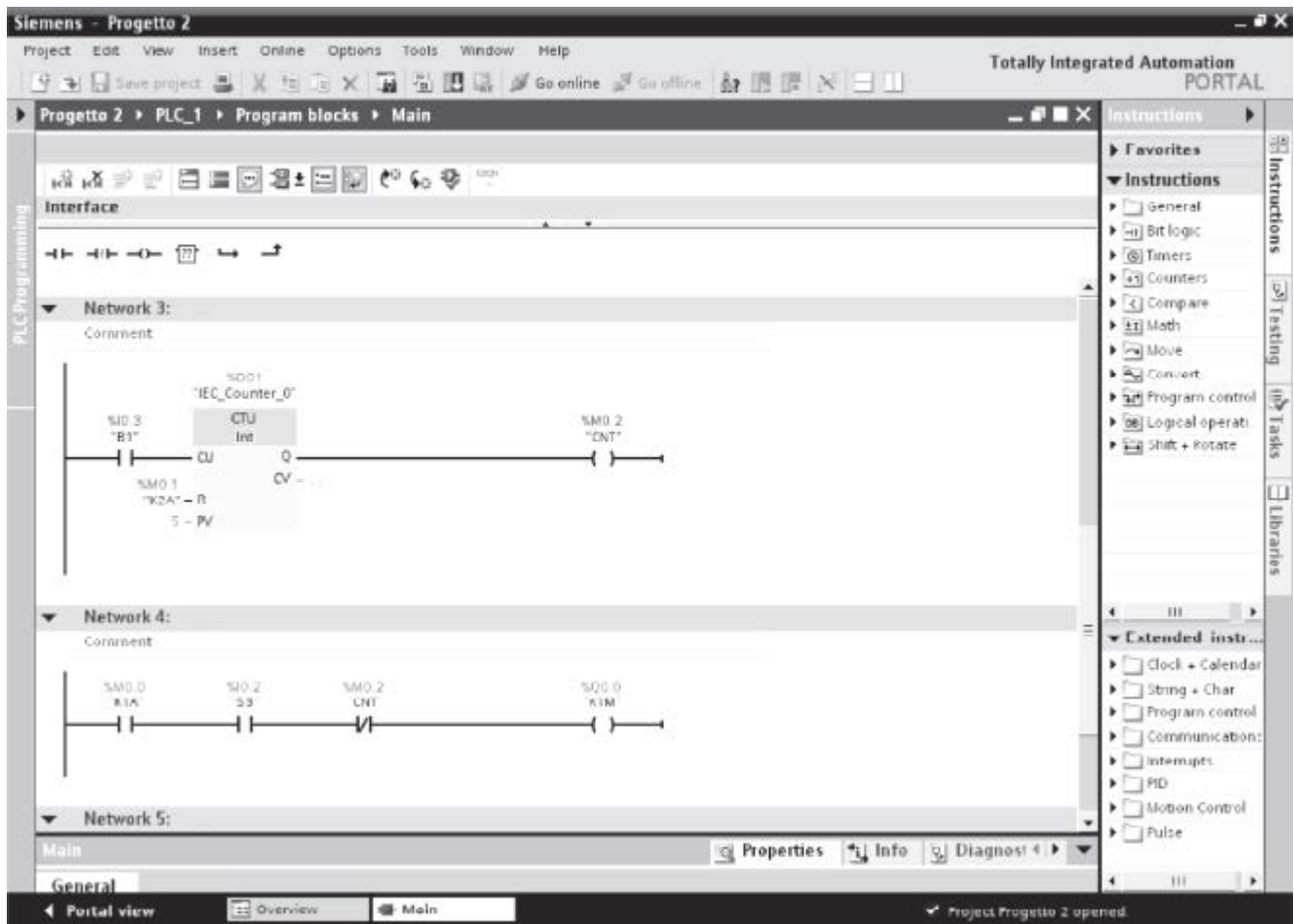


FIGURA 5.5 Editor de programação Ladder.

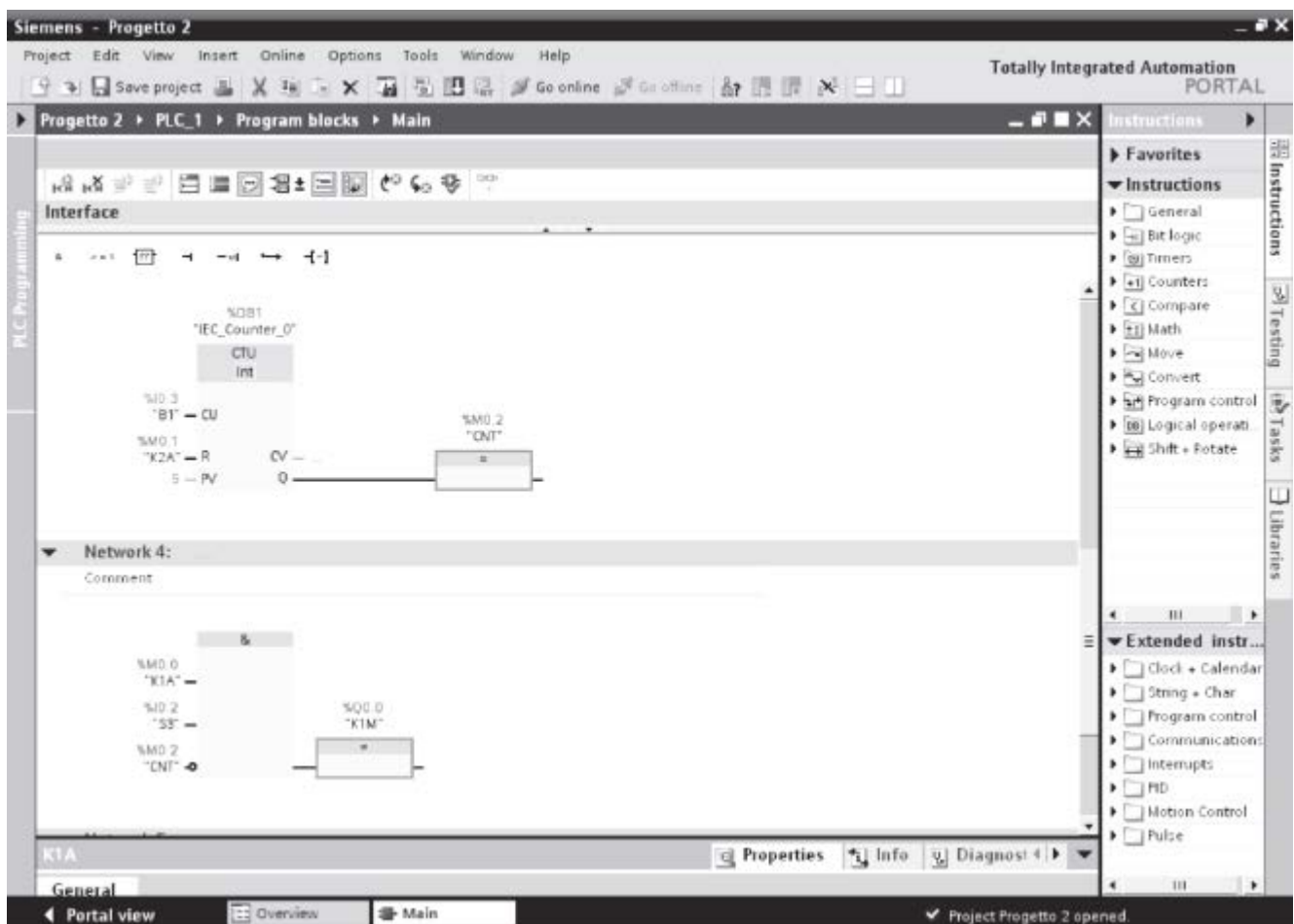


FIGURA 5.6 Editor de programação FBD.

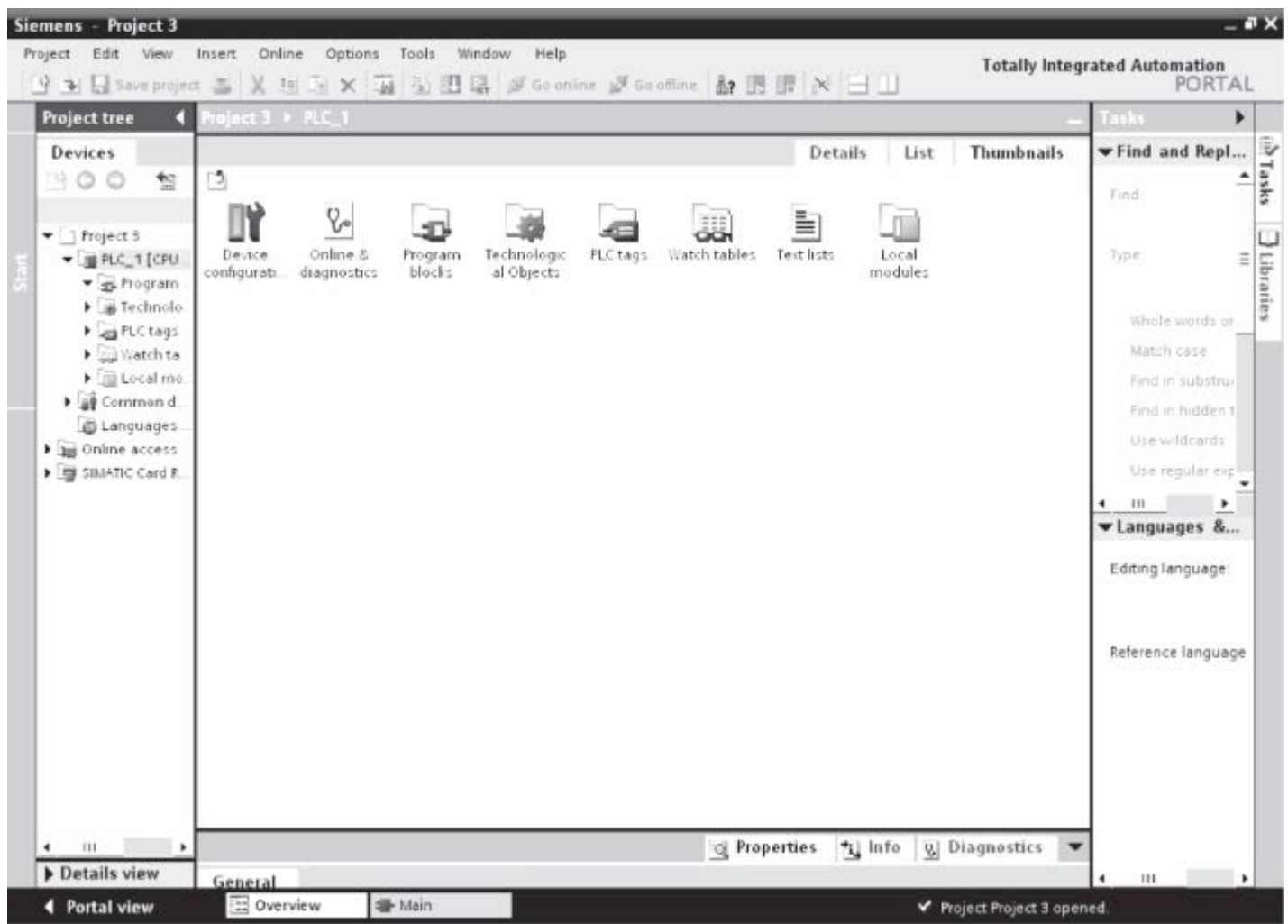


FIGURA 5.7

Para entrar na Tabela dos símbolos, clicar sobre o ícone “PLC tags”, conforme visualizado na Figura 5.7; logo aparece a janela da Figura 5.8.

A Tabela dos símbolos é absolutamente necessária quando o nosso programa é bastante complexo, simplificando muito a identificação de cada variável no sistema de automação.

Quando for necessário, utilizaremos a Tabela dos símbolos conforme Tabela 5.1, que é praticamente a mesma tabela da Figura 5.8.

TABELA 5.1 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	%I0.0	Botão de start
S2	%I0.1	Botão de stop
KM	%Q0.0	Saída contator

5.8 Criação de um Projeto com STEP 7 Basic

Para criar qualquer projeto no STEP 7 Basic é preciso passar pelas seguintes etapas fundamentais:

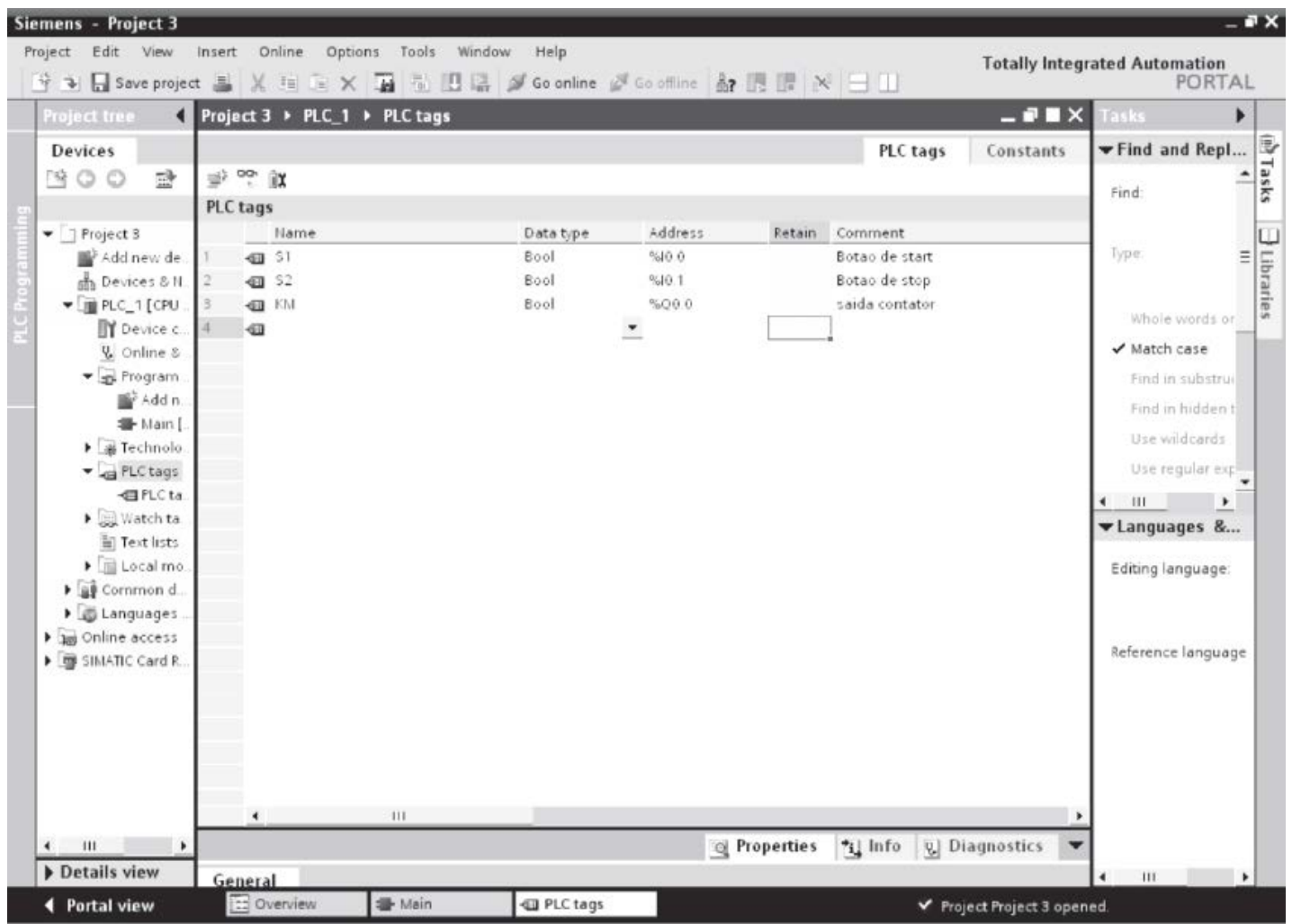


FIGURA 5.8

1. Configurar os dispositivos (Configure a device)

Com STEP 7 Basic é preciso escolher o tipo de CPU utilizada e os tipos e as características dos eventuais módulos adicionais.

2. Criar o programa no PLC (Create a PLC program)

Para escrever o programa na linguagem disponibilizada, no nosso caso o tipo LAD e FBD.

3. Configurar os dispositivos HMI (Configure an HMI screen)

Configurar o painel operador para visualizar o processo ou inserir mensagem para ser visualizada no decorrer do processo a ser controlado, se estiver previsto um dispositivo HMI.

5.9 Exemplo de Criação de um Projeto

Abrindo o STEP 7 Basic logo aparece o Portal view, conforme a Figura 5.9; clicar com o mouse em Create new project. À direita, inserir o nome e o autor do projeto. Confirmar com “Create”.

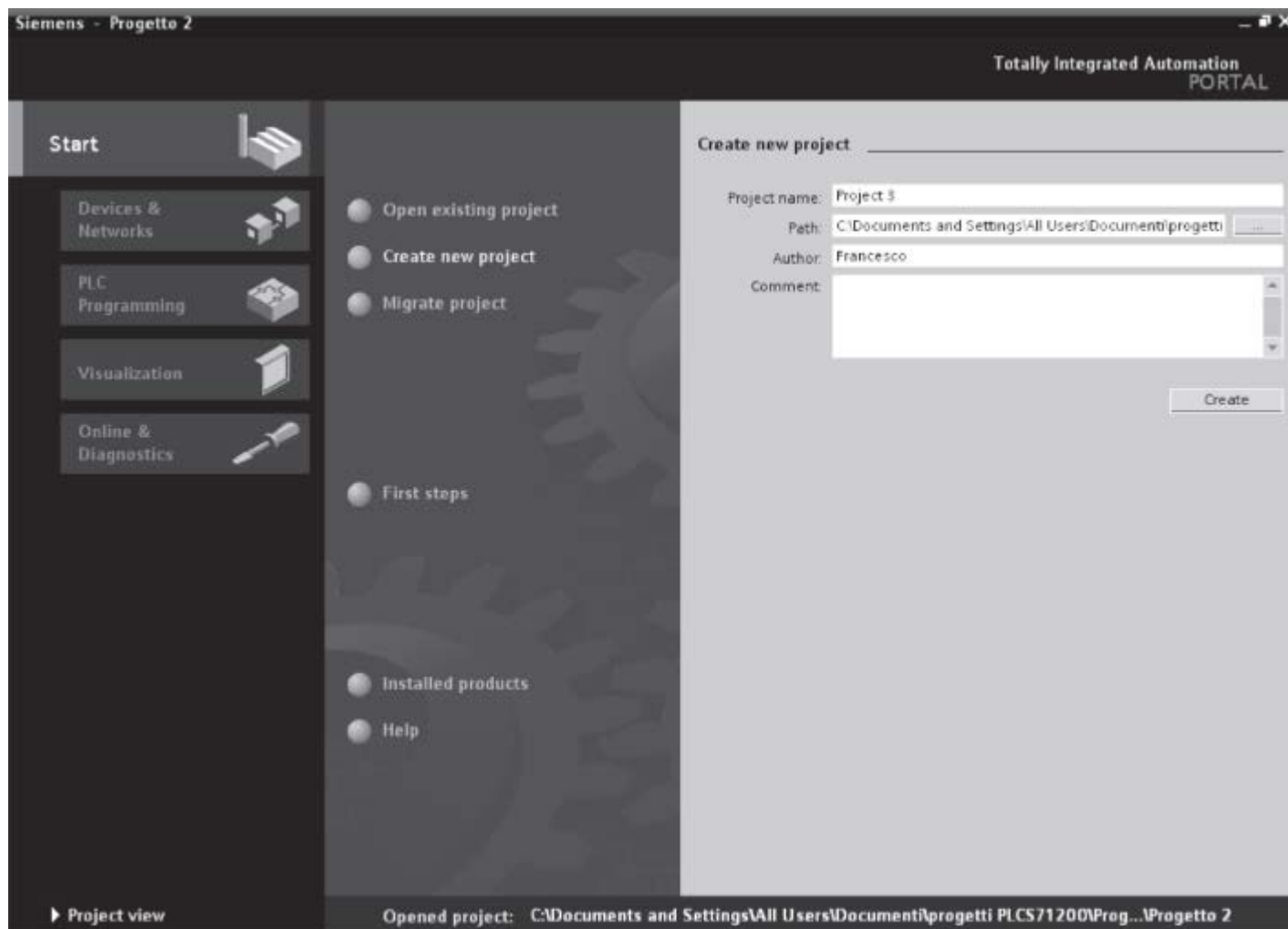


FIGURA 5.9

Logo aparece o Portal view da Figura 5.10; clicar à direita em “Devices & Networks – Configure a device”.

Em seguida aparece a janela da Figura 5.11. Clicar com o mouse em “Add new device” e selecionar à direita a CPU desejada. Supondo que tenha sido selecionada a CPU 1212C AC/DC/RLY com código de ordenação 6ES7-212-1BD30-0XB0, logo aparece a janela conforme a Figura 5.12.

Na Figura 5.12 aparece a CPU 1212C AC/DC/RLY com código de ordenação 6ES7-212-1BD30-0XB0 na imagem digitalizada real.

Abaixo de “Device overview” estão representados os tipos de dispositivos e os respectivos endereços de entrada e saída I/O, código de ordenação, posição no slot, release (firmware) e/ou comentário.

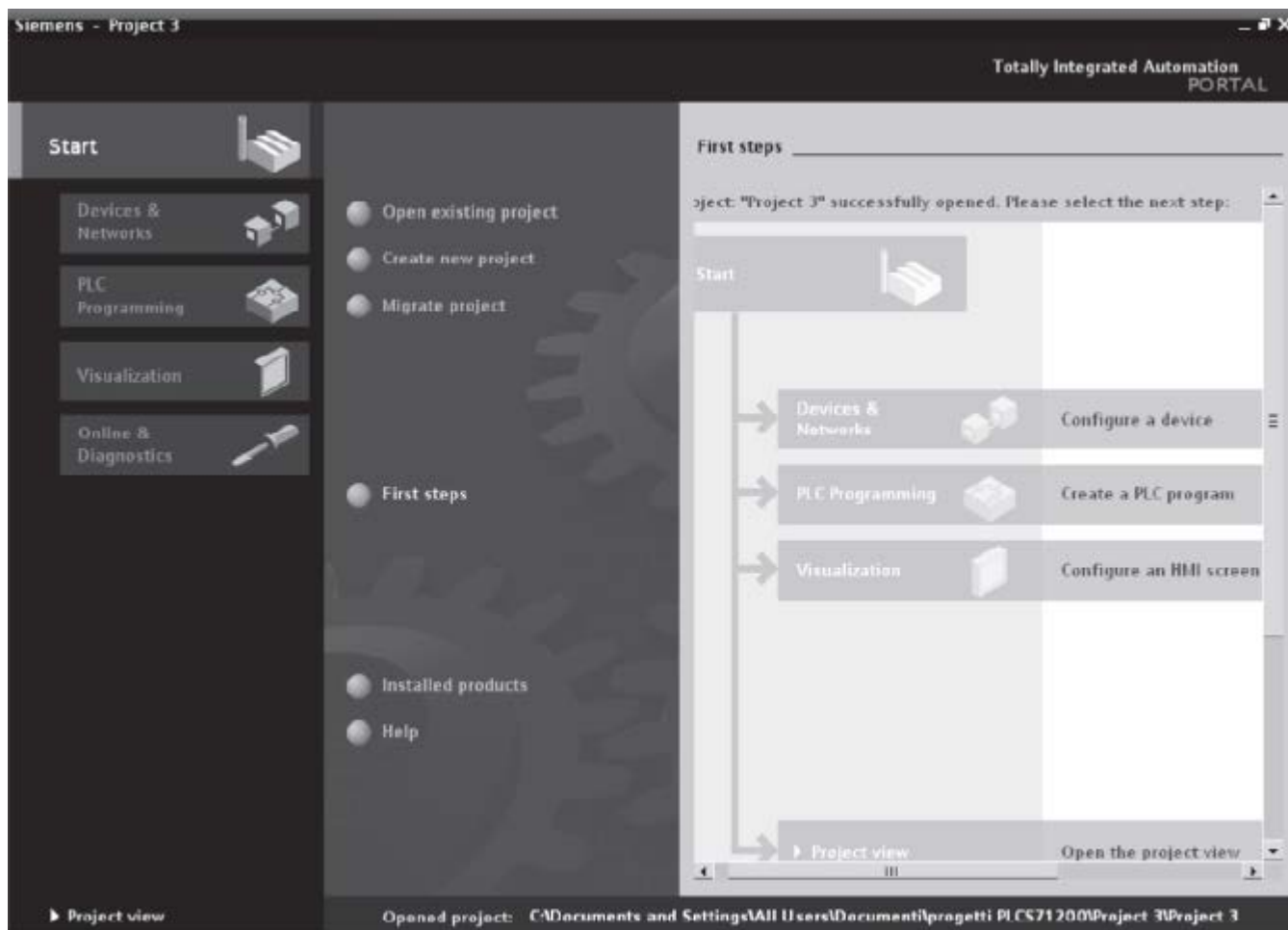


FIGURA 5.10

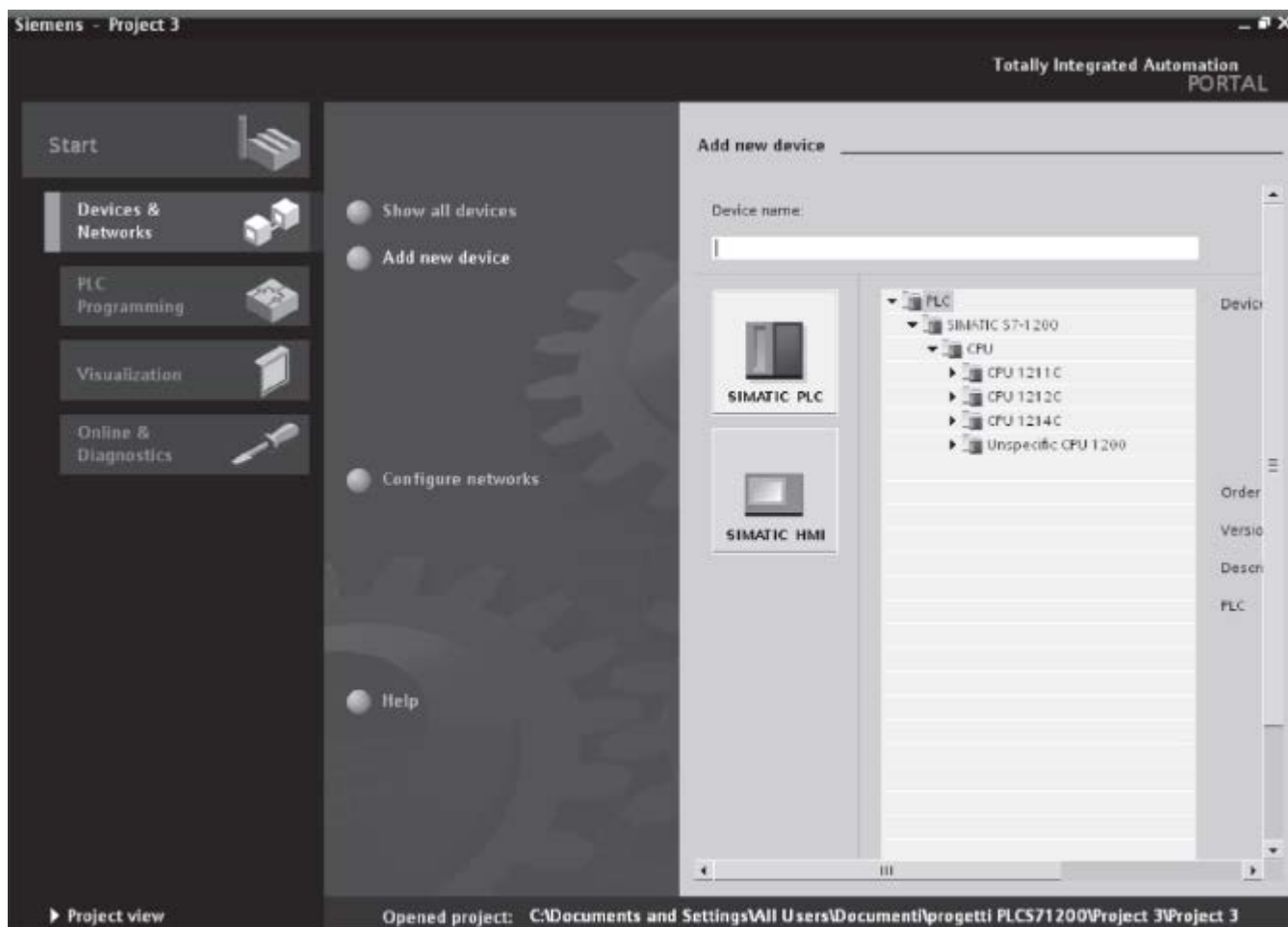


FIGURA 5.11

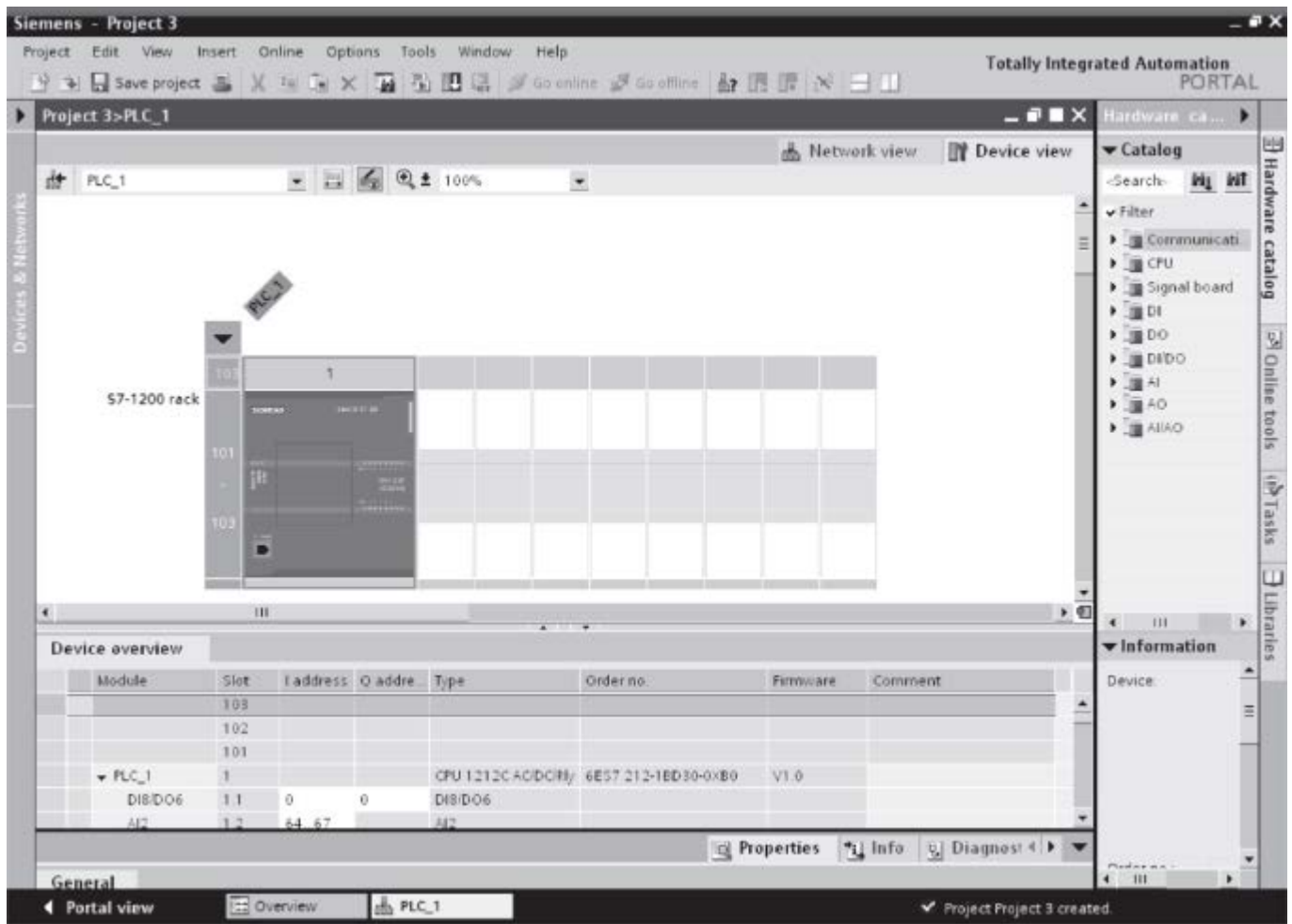


FIGURA 5.12

À direita temos o catálogo Hardware, no qual podemos selecionar os módulos adicionais com um clique do mouse e utilizando a função “Drag & drop”.

Antes de sair do portal de configuração “Dispositivos”, é importante armazenar e compilar a configuração com um clique do mouse nas barras das ferramentas utilizando o ícone

Em seguida é preciso escrever o programa no PLC. Voltamos então ao Portal view, conforme a Figura 5.13, e clicamos com o mouse em “PLC programming”. Logo aparece a janela da Figura 5.14.

Clicando à direita sobre o bloco Main , logo aparece a janela do Project view, conforme a Figura 5.15. O editor Project view é selecionado para a linguagem Ladder.

Seleciona-se o componente tipo contato normalmente aberto, fechado, bobina etc. utilizando o bloco . Com a função Drag & drop, insere-se o componente na linha de programa conforme a Figura 5.16.

A linha de programa completa é a que se vê na Figura 5.17. Sobre cada contato ou bobina escreve-se, no caso de entrada do PLC, %I0.0, %I0.1 e assim sucessivamente para outras eventuais entradas. No caso de saídas, temos %Q0.0, %Q0.1 e assim sucessivamente para outras eventuais saídas.

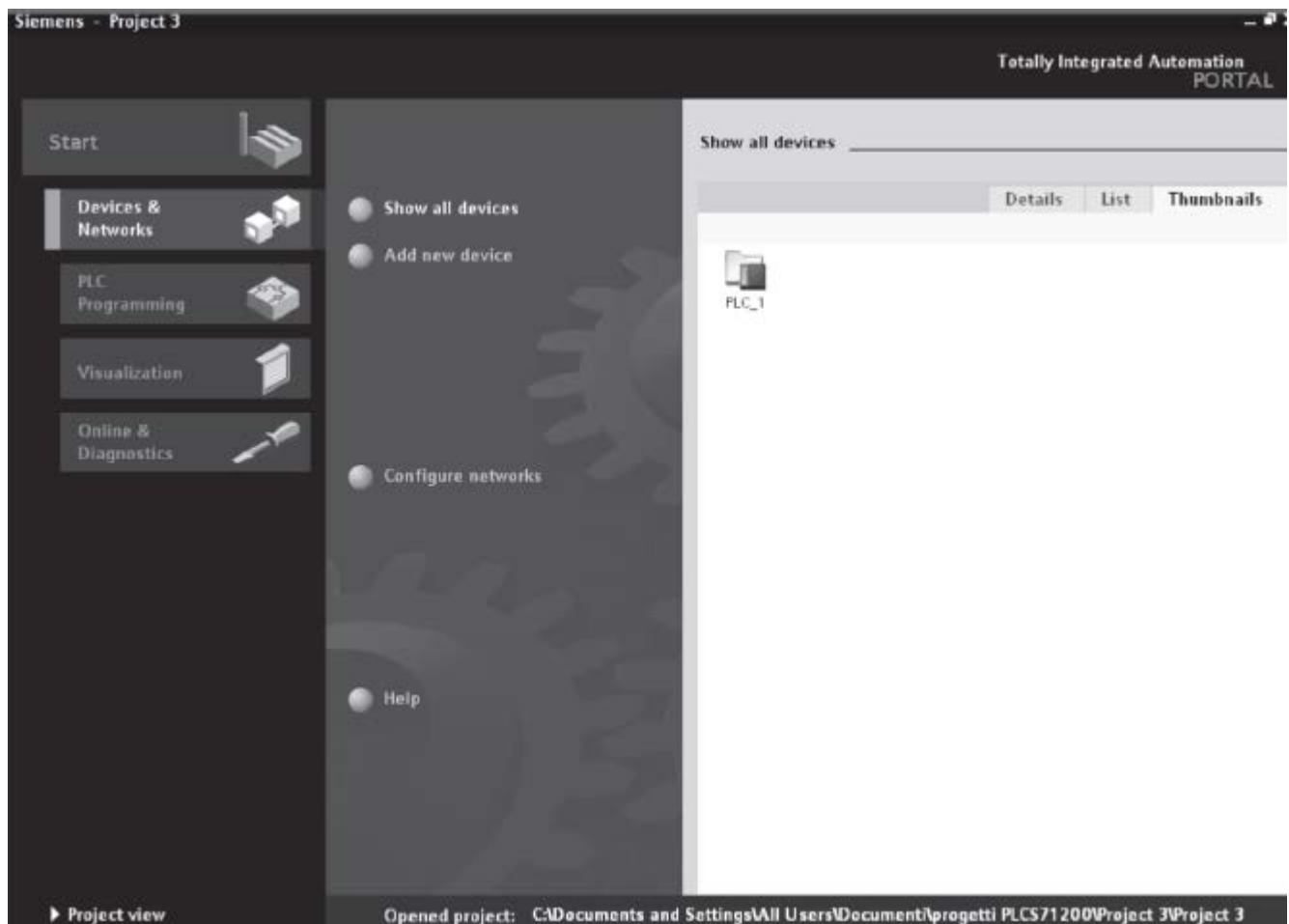


FIGURA 5.13

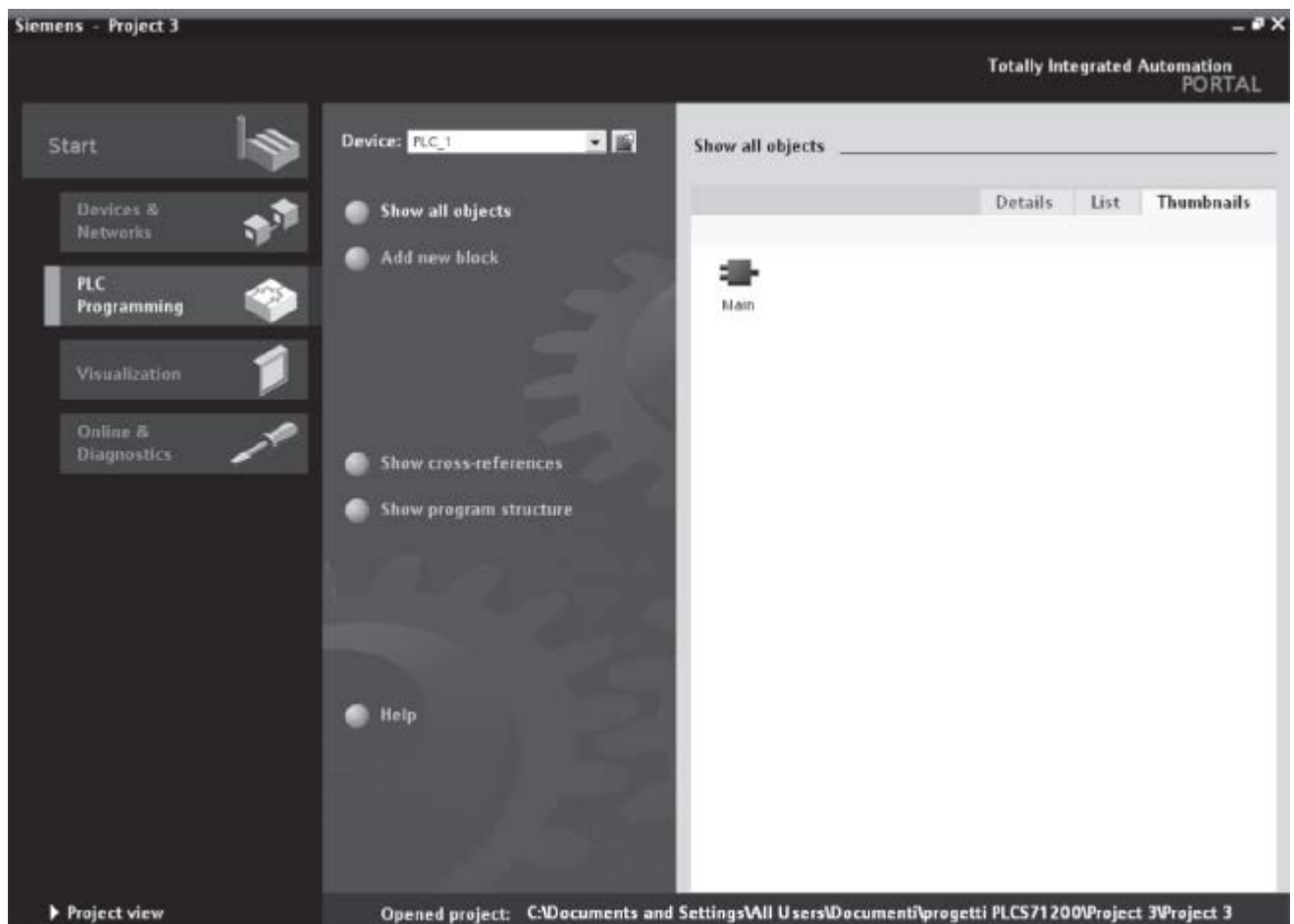


FIGURA 5.14

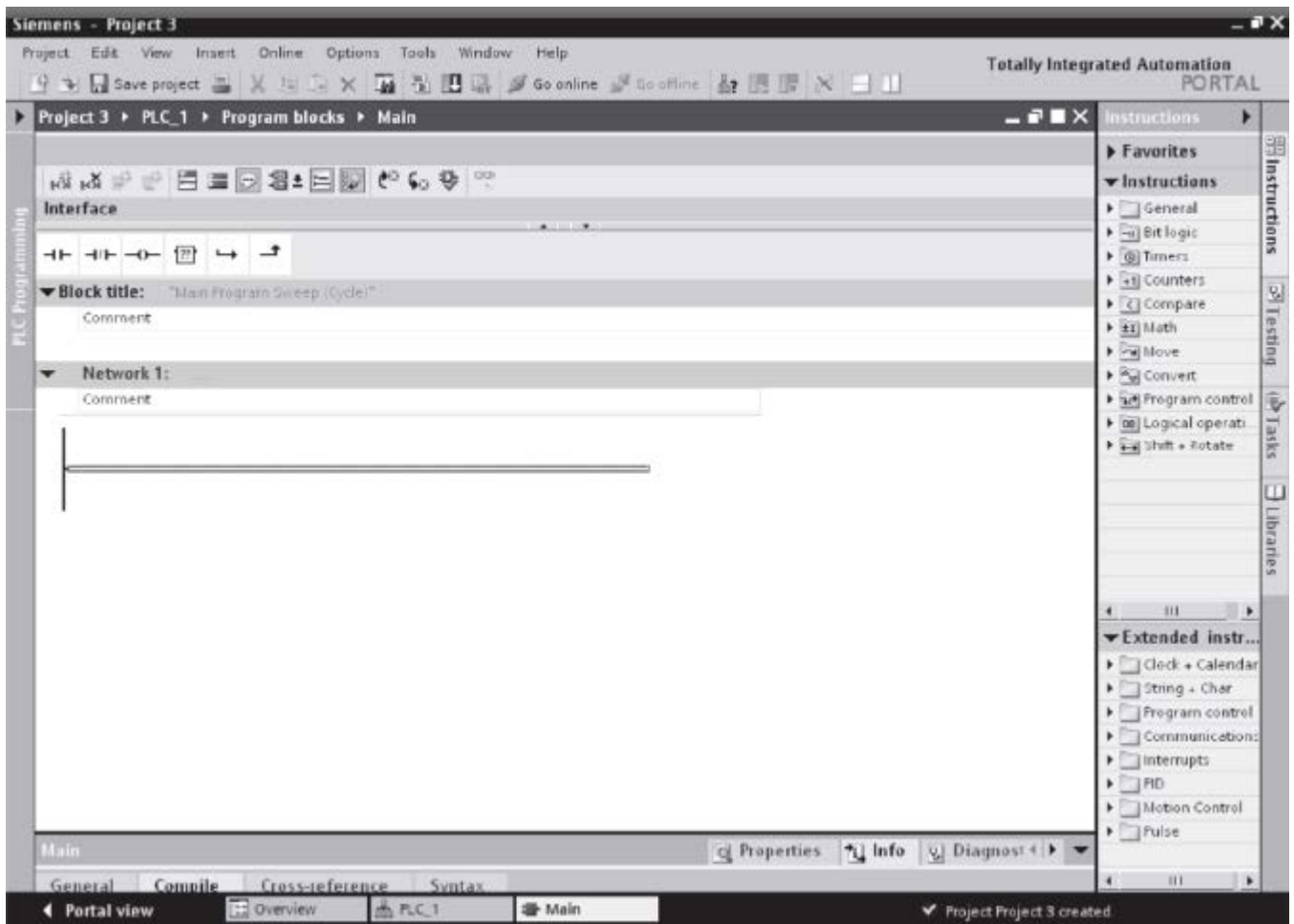


FIGURA 5.15

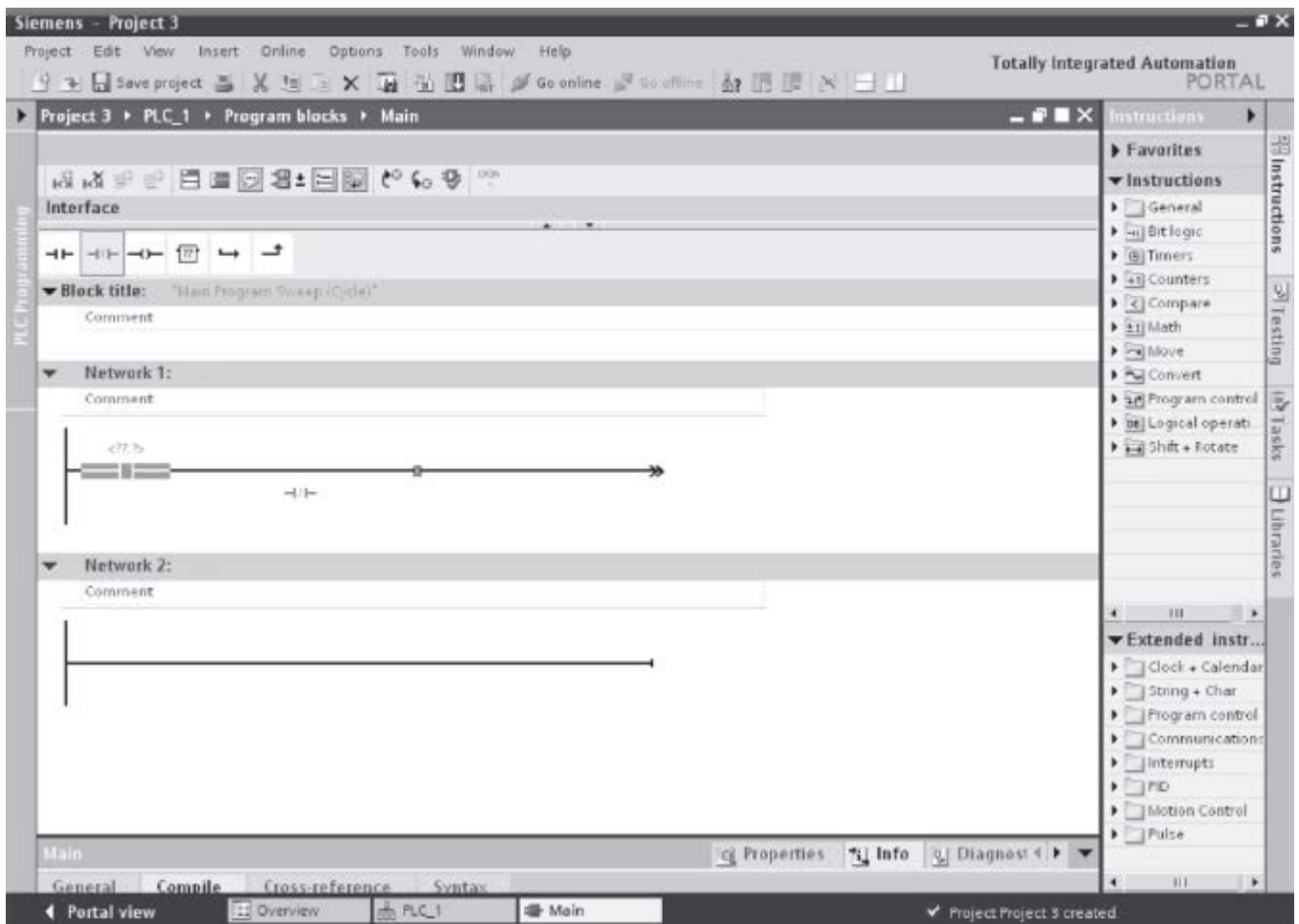


FIGURA 5.16

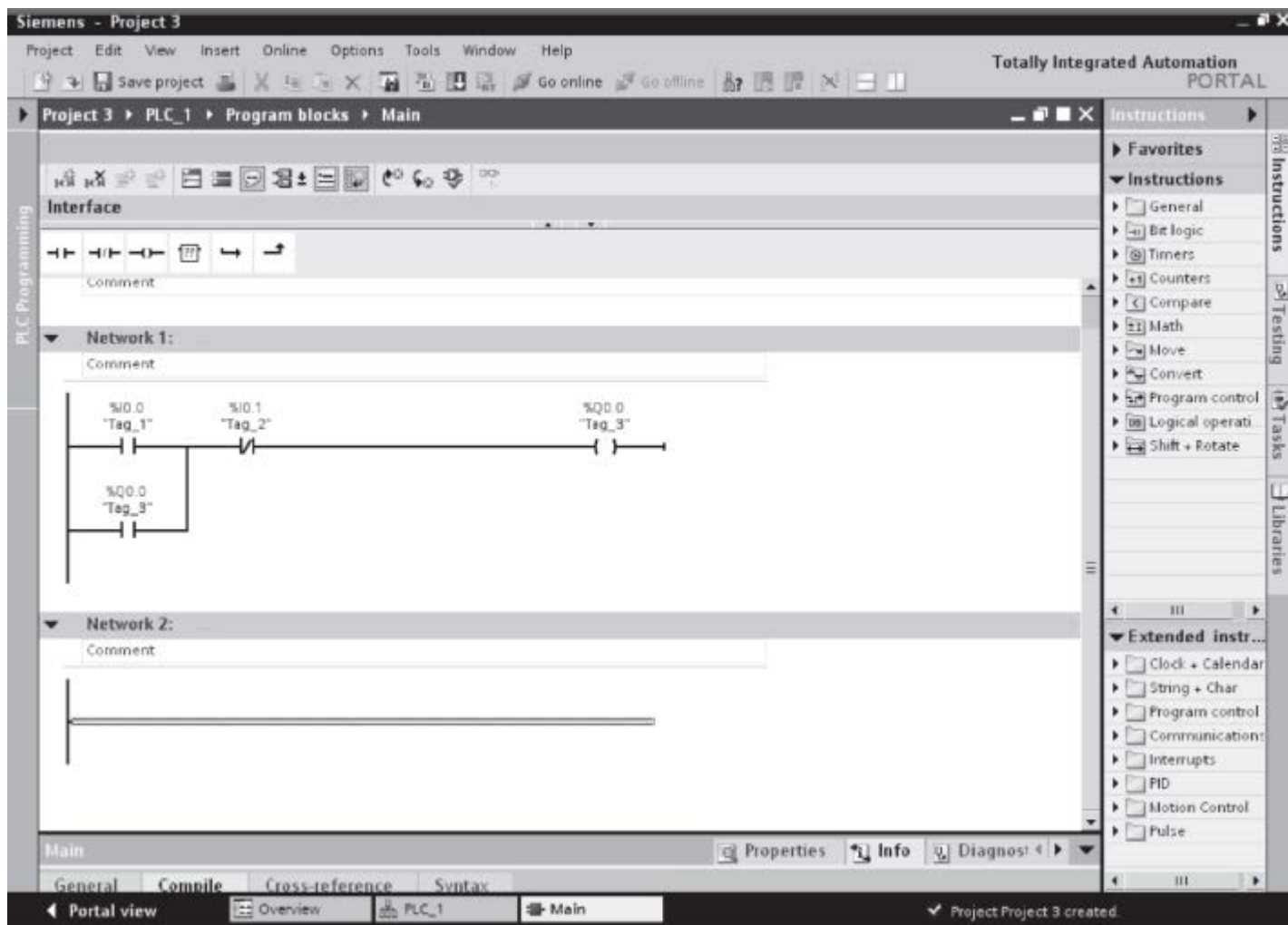




FIGURA 5.17

Sobre cada contato e bobina aparece automaticamente a sigla tag_1, tag_2 etc. De fato, o PLC interpreta a entrada e a saída como uma variável para ser sucessivamente elaborada, por exemplo, em um dispositivo HMI.

Antes de sair do portal de configuração “Dispositivos”, é importante armazenar e compilar a configuração com um clique do mouse nas barras das ferramentas utilizando o ícone .

Em seguida, Salvar o projeto .

Para converter o programa Ladder no equivalente FBD, clicar embaixo de “Overview”. Logo aparece a janela da Figura 5.18.

Ao clicar com o botão direito sobre o bloco , escolha o item “Properties”; aparece assim a propriedade do bloco, sendo então possível a escolha da linguagem de programação, no nosso caso FBD. Confirmar com “OK”.

Logo aparece o editor da linguagem de programação FBD, conforme a Figura 5.19.

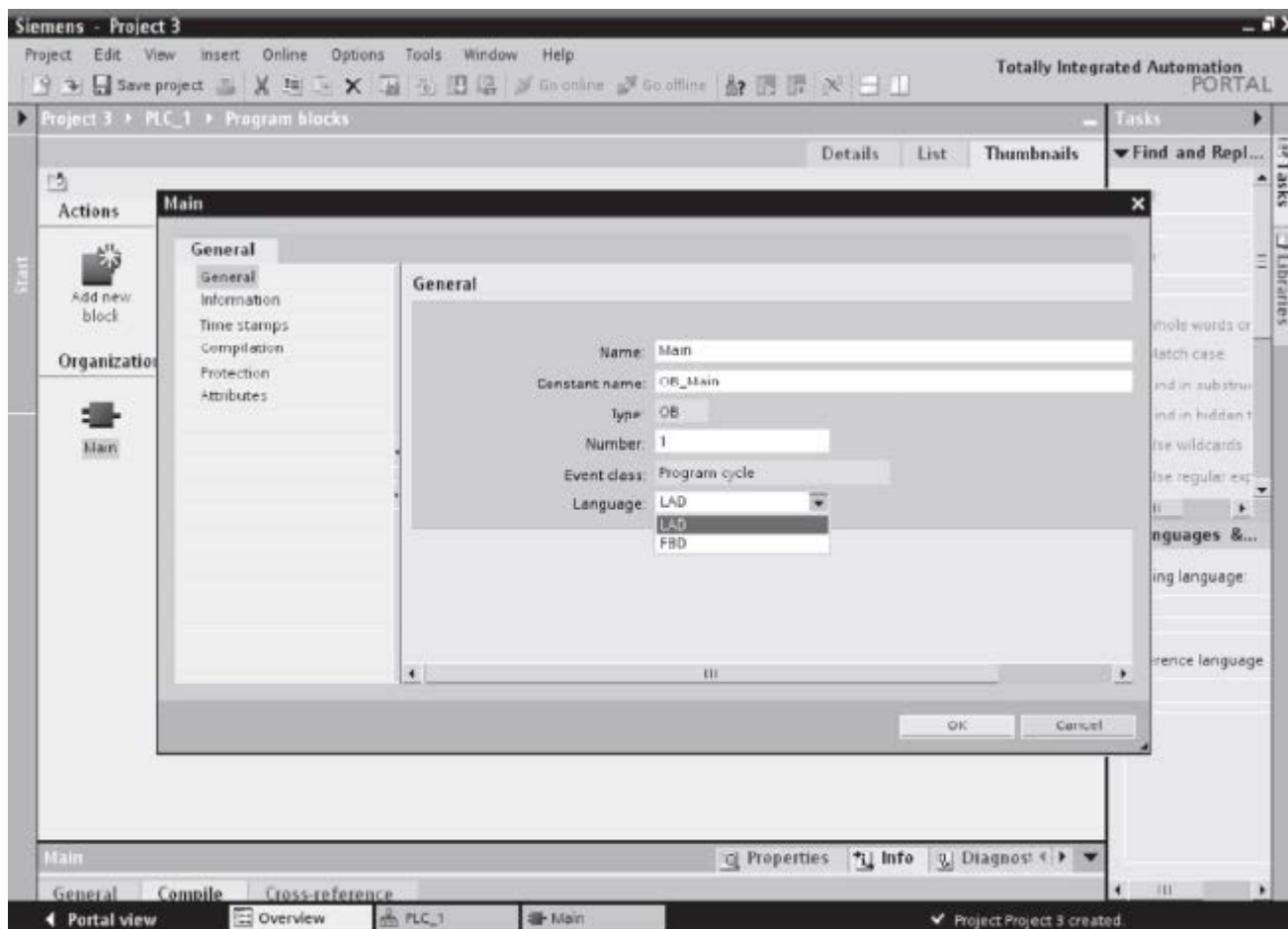


FIGURA 5.18

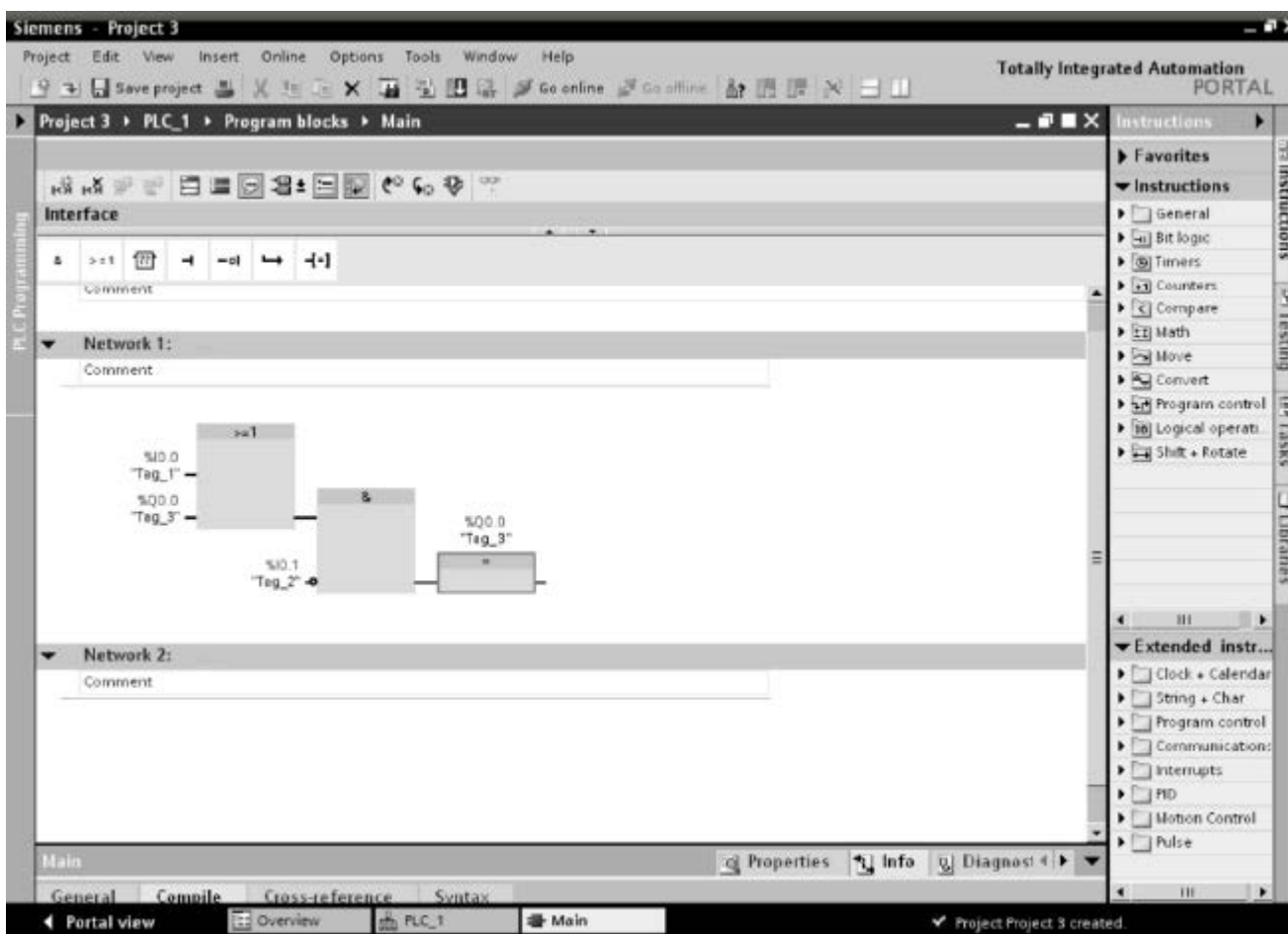


FIGURA 5.19

5.9.1 Configurar a Porta de Comunicação

Antes de carregarmos a configuração da CPU, o endereço IP da porta PROFINET precisa ser definido. O mesmo STEP 7 Basic sugere um endereço padrão, mas podemos também defini-lo manualmente.

Para executar essa tarefa, volta-se logo na configuração Hardware e clica-se com o botão direito no item “Propriedade”, na janelinha embaixo, conforme sugerido na Figura 5.20.

Logo aparece a janela da configuração da porta PROFINET, com o endereço IP do STEP 7 Basic criado por nós (veja a Figura 5.21).

Esse exemplo usa o endereço IP 192.168.0.1, com máscara de rede 255.255.255.0. Conforme nossas necessidades, pode ser preciso configurar um endereço IP diferente para a CPU.

Para mais informações, consultar em cada caso o manual do sistema.

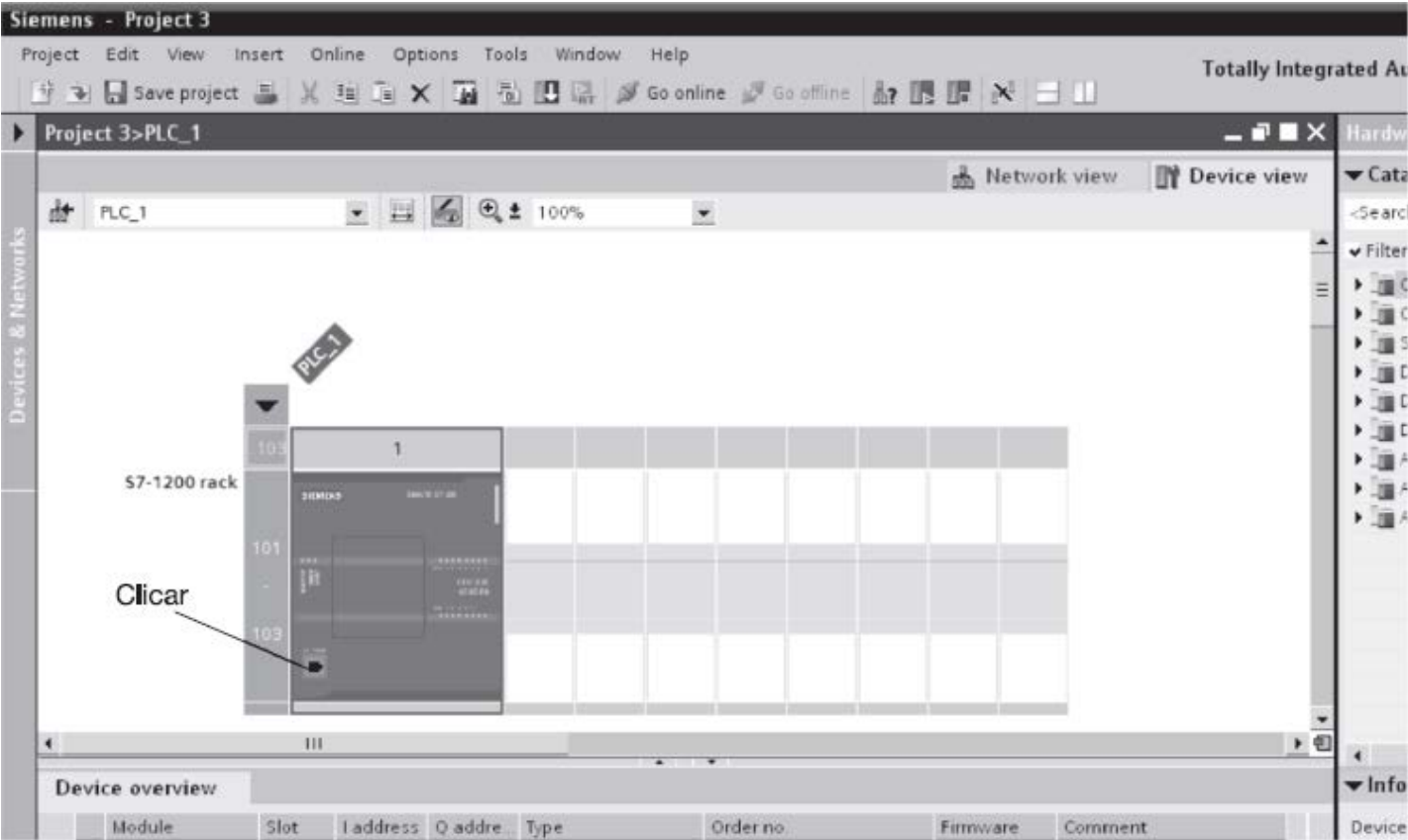


FIGURA 5.20

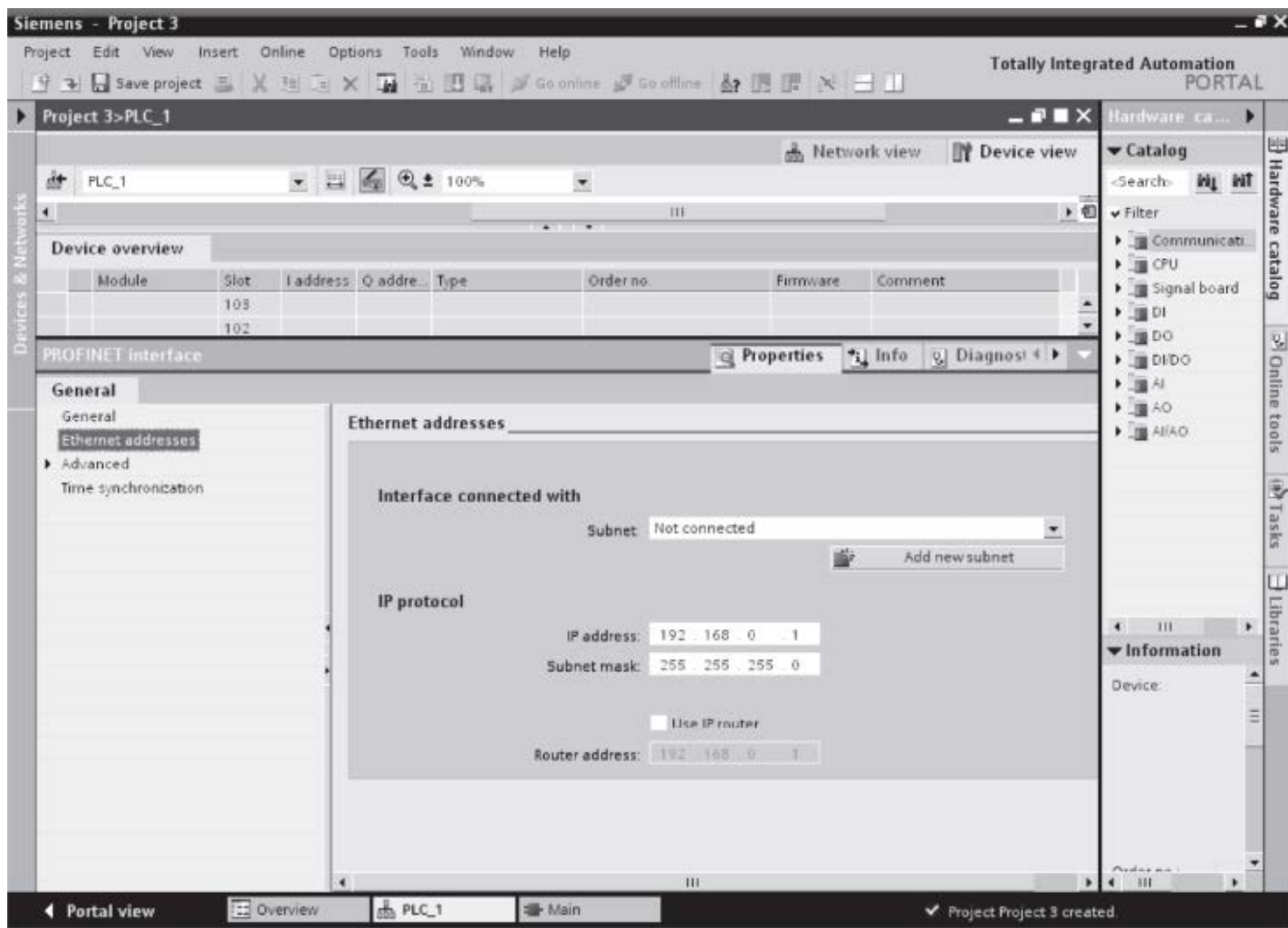


FIGURA 5.21

5.9.2 Ligação dos Dispositivos

Antes de executar o download do nosso projeto do computador para o PLC, precisamos ligar o cabo PROFINET do computador ao PLC, como sugerido na Figura 5.22.

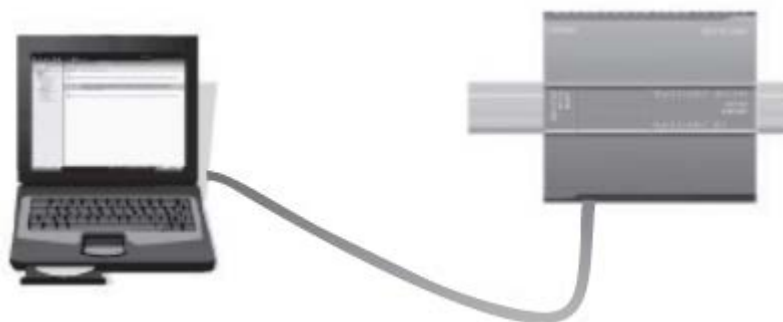


FIGURA 5.22

Após inserir o simulador nos parafusos de entrada da CPU 1212C AC/DC/RLY, alimentar a CPU a 120 ou 230 VAC, conforme a Figura 5.23.

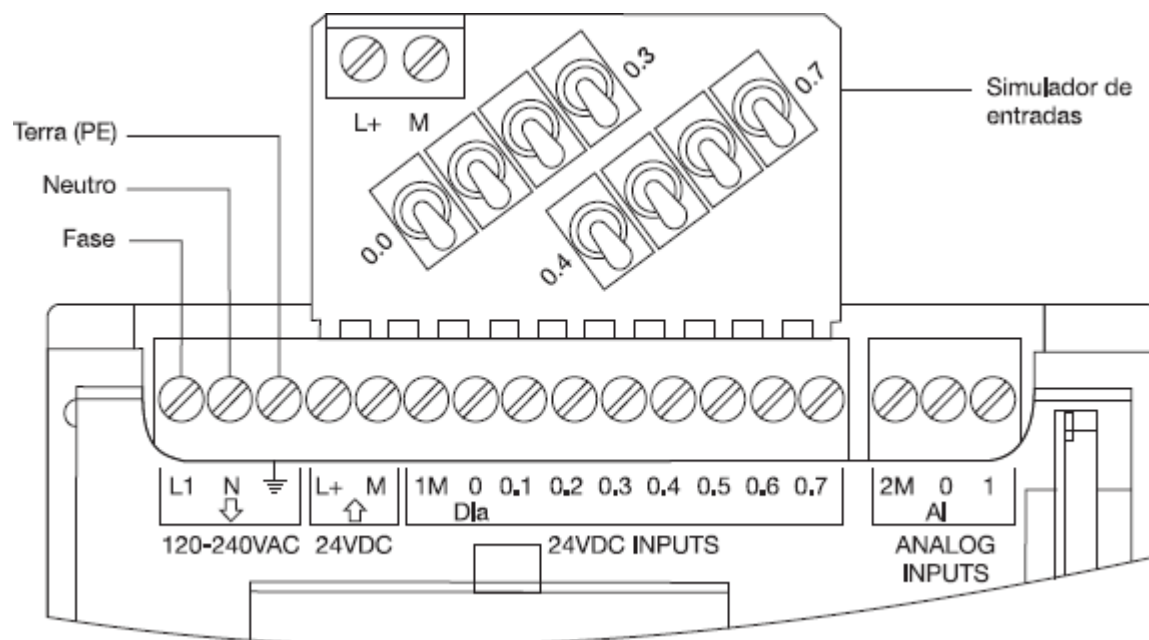





FIGURA 5.23

5.9.3 Download e Simulação do Programa

Nesta fase precisamos carregar o nosso projeto do computador na CPU; então voltamos ao editor de programação Ladder (veja a Figura 5.24), na barra da ferramenta, é preciso clicar com o mouse no ícone de compilação  e no ícone de carregamento (download)  do computador para o PLC.

Logo aparece a janela de download, conforme a Figura 5.25. Confirmar “Load”.

Para o teste prático é preciso comutar a chave em teste (do simulador de entradas) e controlar a energização da saída correspondente (veja a Figura 5.26).

Clicando com o mouse no ícone , logo aparece a simulação do programa em tempo real, conforme a Figura 5.28.

A janela da Figura 5.27 permite comutar a CPU no estado RUN, STOP, MRES, já discutido na Seção 3.1.3, Capítulo 3.

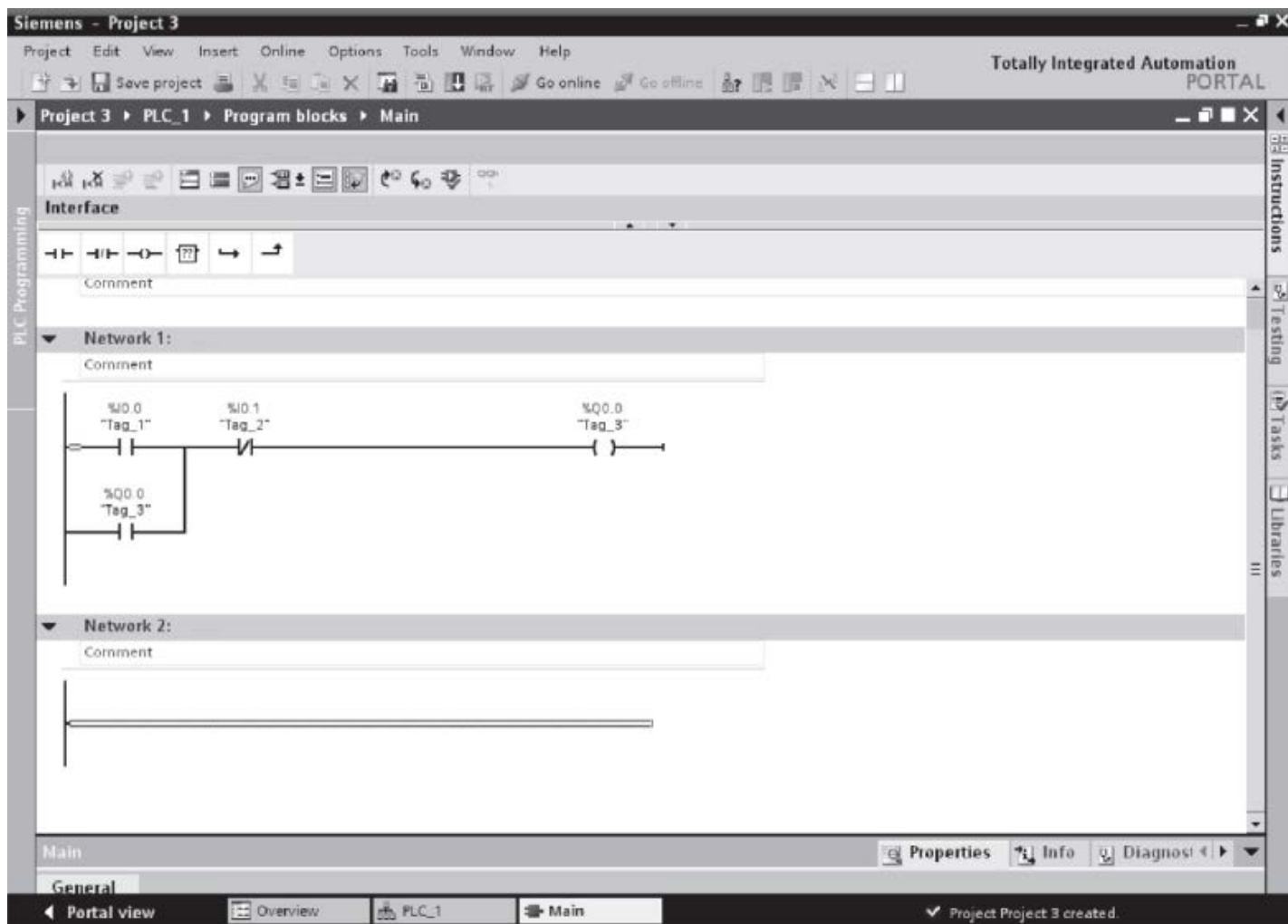


FIGURA 5.24

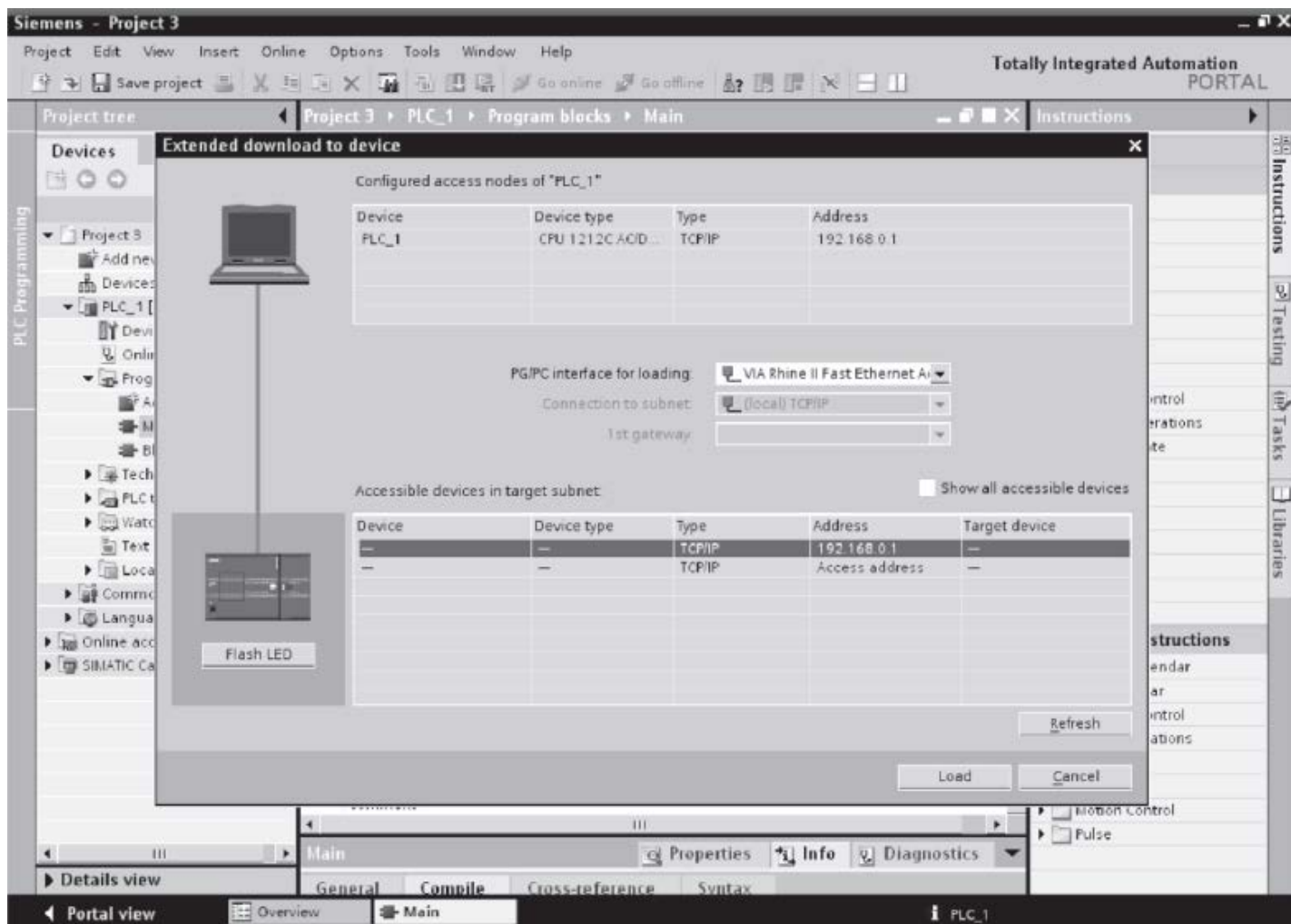


FIGURA 5.25



FIGURA 5.26

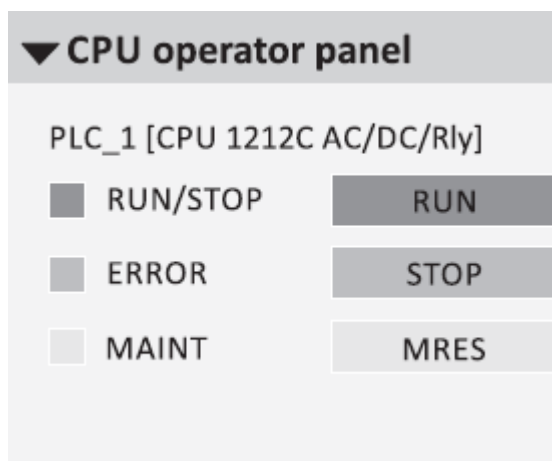


FIGURA 5.27

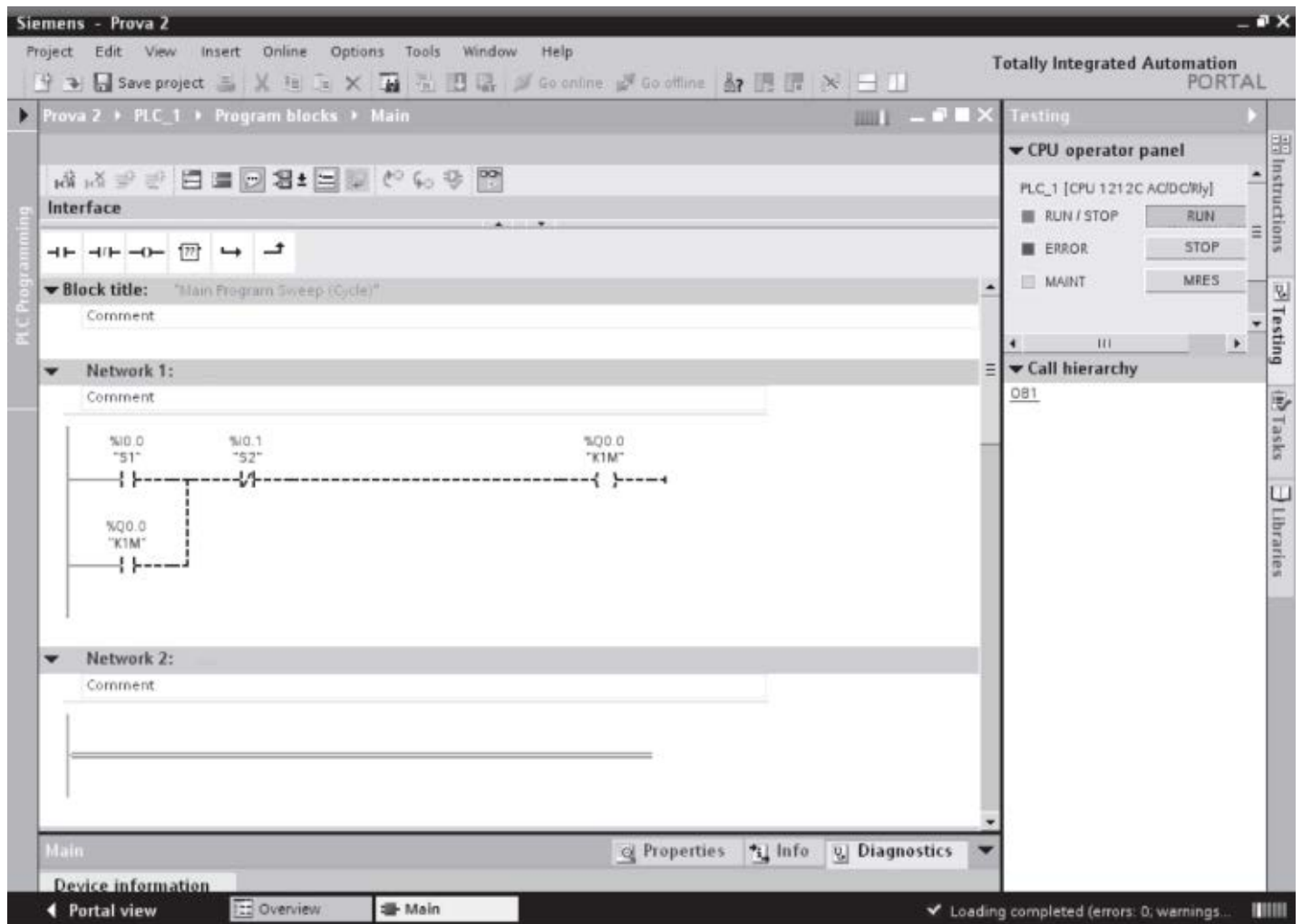


FIGURA 5.28

CARACTERÍSTICAS DO SOFTWARE DO PLC S7-1200

CARACTERÍSTICAS DO SOFTWARE DO PLC S7-1200

- 6.0 Generalidades
 - 6.1 Bit, Byte, Word
 - 6.2 Tipos de Dados na CPU S7-1200
 - 6.3 Endereçamento dos Dados
 - 6.4 Modalidade de Endereçamento dos Principais Áreas/Dados do PLC S7-1200
 - 6.5 Configuração dos Endereços com STEP 7 Basic
 - 6.6 A Programação de Blocos
 - 6.7 Programação Linear e Estruturada
 - 6.8 Exemplo de um DB Instance e DB Global
 - 6.9 Blocos com Parâmetros
-

6.0 Generalidades

Nos dispositivos digitais, a unidade mínima de informação é chamada de *bit*; cada bit armazena somente dois estados possíveis: 0 ou 1.

Por exemplo, a chave de ligação da luz pode estar aberta ou fechada.

O PLC S7-1200 é um dispositivo digital com a arquitetura interna de 32 bits, altamente sofisticado e com um set de instruções muito vasto.

6.1 Bit, Byte, Word

A CPU organiza os bits de dados em grupos. Cada grupo de 8 bits é chamado de byte. Veja a Figura 6.1.

CARACTERÍSTICAS DO SOFTWARE DO PLC S7-1200

- 6.0 Generalidades
- 6.1 Bit, Byte, Word
- 6.2 Tipos de Dados na CPU S7-1200
- 6.3 Endereçamento dos Dados
- 6.4 Modalidade de Endereçamento dos Principais Áreas/Dados do PLC S7-1200
- 6.5 Configuração dos Endereços com STEP 7 Basic
- 6.6 A Programação de Blocos
- 6.7 Programação Linear e Estruturada
- 6.8 Exemplo de um DB Instance e DB Global
- 6.9 Blocos com Parâmetros

6.0 Generalidades

Nos dispositivos digitais, a unidade mínima de informação é chamada de *bit*; cada bit armazena somente dois estados possíveis: 0 ou 1.

Por exemplo, a chave de ligação da luz pode estar aberta ou fechada.

O PLC S7-1200 é um dispositivo digital com a arquitetura interna de 32 bits, altamente sofisticado e com um set de instruções muito vasto.

6.1 Bit, Byte, Word

A CPU organiza os bits de dados em grupos. Cada grupo de 8 bits é chamado de byte. Veja a Figura 6.1.

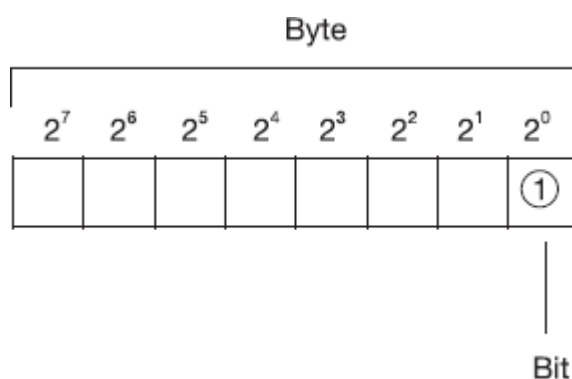


FIGURA 6.1

Cada bit do grupo é definido a partir de uma posição com um endereço próprio numerado de 0 a 7.

Um grupo de 2 bytes é chamado de *palavra* (word), e um grupo de 4 bytes é chamado de *dupla palavra* (double word). Veja a Figura 6.2.

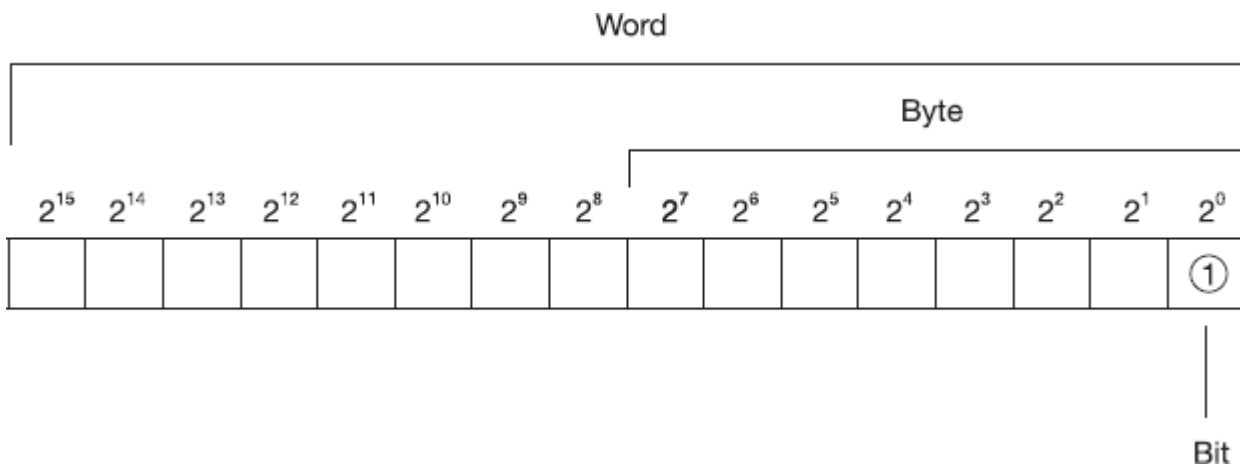


FIGURA 6.2

Para os números utiliza-se o sistema binário (com base 2). Uma word pode representar um número compreendido entre -32768 e $+32767$. O bit com valor 2^{15} é usado para definir um número negativo (se a posição 2^{15} possui o valor 1, o número é negativo, do contrário é positivo, valor 0).

A CPU S7-1200 suporta vários grupos de bytes, até formar um tipo de dados muito grande, tipo o *Long real*, número real longo, para o armazenamento de valores elevados e muito precisos.

Por exemplo, o campo de dados do Lreal vai de $\pm 2,23 \times 10^{-308}$ a $\pm 1,79 \times 10^{-308}$.

6.2 Tipos de Dados na CPU S7-1200

Os tipos de dados são específicos ao tamanho e à estrutura interna dos bits.

Nas Tabelas 6.1 e 6.2 temos um resumo de todos os tipos de dados das CPUs S7-1200, rigorosamente conforme a norma IEC 61131-3.

6.3 Endereçamento dos Dados

As CPUs S7-1200 armazenam as informações em diferentes posições da memória que têm endereço preestabelecido. O programador poderá acessar diretamente essa informação especificando a área de memória à qual queira ter acesso e o seu endereço.

A Figura 6.3 mostra o modo de acesso a um bit; nesse exemplo, a área de memória e o endereço do byte (M = área Merker e 3 = byte 3) são seguidos de um ponto decimal (“.”) que separa o endereço do bit (bit 4).

TABELA 6.1

Tipos de dados	Tamanho	Campo	Exemplo de constante
Bool	1 bit	De 0 a 1	TRUE, FALSE, 0, 1
Byte	8 bit (1 byte)	De 16#00 a 16#FF	16#12, 16#AB
Word	16 bit (2 byte)	De 16#0000 a 16#FFFF	16#ABCD, 16#0001
DWord	32 bit (4 byte)	De 16#00000000 a 16#FFFFFFFF	16#02468ACE
Char	8 bit (1 byte)	De 16#00 a 16#FF	“A”, “t”, “@”
SInt	8 bit (1 byte)	De -128 a 127	123, -123
USInt	8 bit (1 byte)	De 0 a 255	123
Int	16 bit (2 byte)	De -32.768 a 32.767	123, -123
UInt	16 bit (2 byte)	De 0 a 65.535	123
DInt	32 bit (4 byte)	De -2.147.483.648 a 2.147.483.647	123, -123
UDInt	32 bit (4 byte)	De 0 a 4.294.967.295	123

TABELA 6.2

Tipos de dados	Tamanho	Campo	Exemplo de constante
Real	32 bit (4 byte)	De $\pm 1, 18 \times 10^{-38}$ a $\pm 3,40 \times 10^{38}$	123, 456; -3,4; -1,2E + 12; 3,4E-3
Lreal	64 bit (8 byte)	De $\pm 2,23 \times 10^{-308}$ a $\pm 1,79 \times 10^{308}$	12345.123456789 -1,2E + 40
Time	32 bit (4 byte)	T#24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Armazenado como: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#2d T#1d_2h_15m_30x_45ms
String	Variável	De 0 a 254 caracter em formato byte	‘ABC’

DTL1	12 byte	Mínimo DTL#1970-01-01-00:00:00.0 Máximo DTL#2554-12-31-23:59:59.999.999.999	DTL#2008-12-16-20:30:20.250
------	---------	--	-----------------------------

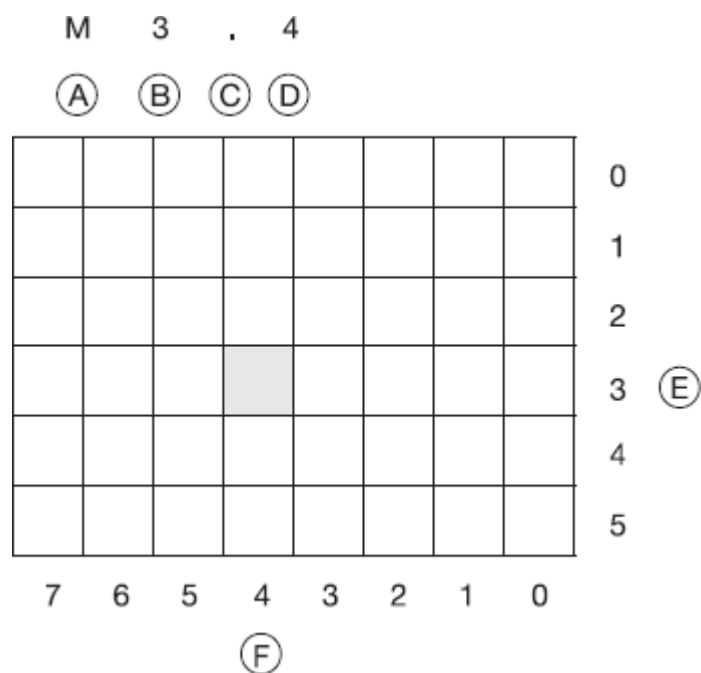


FIGURA 6.3

– Identificador de área

São constituídos de uma letra que identifica os vários elementos do controlador programável (veja a Tabela 6.3).

Os controladores Siemens S7-1200 podem ser programados utilizando a sintaxe das instruções nos idiomas inglês e alemão. No entanto, para que o nosso programa seja conforme a norma IEC 61131-3 é preciso utilizar o inglês.

A sintaxe em alemão é muito utilizada na Europa por não ser conforme a nenhuma norma internacional. Nesta obra se utilizará exclusivamente a sintaxe em inglês, que é conforme a norma internacional IEC 61131-3.

TABELA 6.3

Código elemento em inglês	Descrição
I	Registro de imagem do processo das entradas
Q	Registro de imagens do processo das saídas
M	Área de memória Merker ou relé internos
L	Área de memória temporária
DB	Data block

– Identificador do modo de acesso

É constituído de uma letra que especifica a modalidade com a qual se acessará a área dos dados. Veja a Tabela 6.4.

TABELA 6.4

Código modo	Descrição
B	Acesso a byte
W	Acesso a Word ou palavra (2 bytes consecutivos)
D	Acesso a double Word (4 bytes consecutivos)

– Endereço Byte

O endereço de vários elementos pode variar conforme a CPU utilizada e a configuração de hardware adotada. Assim, a consulta ao manual de sistema é indispensável.

Na Figura 6.4 temos um exemplo de endereço do byte do registro de imagem do processo das entradas IB0.

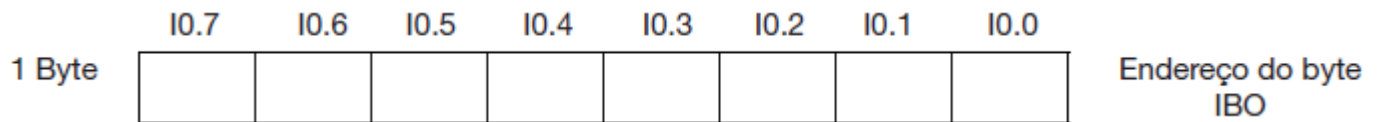


FIGURA 6.4

Na Figura 6.4 temos o registro IB0 (8 bits), composto das entradas I0.0 até I0.7.

– Endereço Word

Na Figura 6.5 temos um exemplo de endereço word do registro de imagem do processo das saídas QW0.

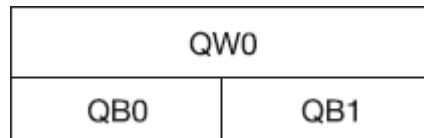


FIGURA 6.5

Na Figura 6.6 temos o registro QW0 (16 bits), composto dos bytes QB0 e QB1.

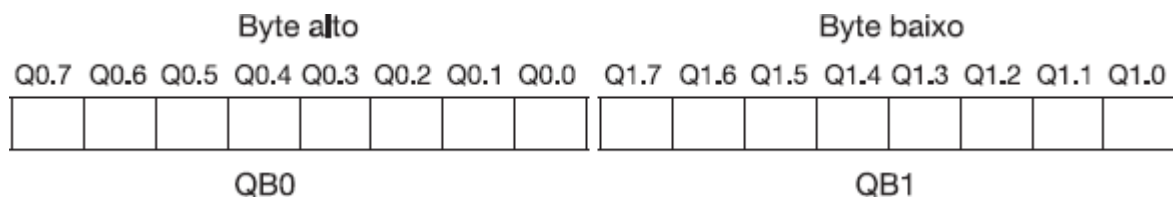


FIGURA 6.6

Na Figura 6.6 temos os bytes QB0 e QB1, com os respectivos bits.

– Endereço Double Word

Na Figura 6.7, temos um exemplo de endereço double word do registro de imagem do processo das entradas ID0.

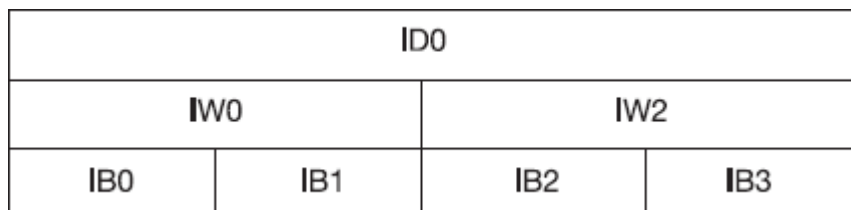


FIGURA 6.7

Na Figura 6.7, temos o registro ID0 (32 bits), composto da double word IW0 (16 bits) e IW2 (16 bits).

6.4 Modalidade de Endereçamento das Principais Áreas/Dados do PLC S7-1200

– Registro de imagens dos processos das entradas (I) e das saídas (Q)

É possível, como já vimos, acrescentar a CPU de outros pontos de I/O conectando as unidades de ampliação, de modo a formar uma sequência de entradas e saídas.

Os endereços das entradas e saídas de ampliação são determinados pelo sistema operacional STEP 7 Basic de forma automática por meio do configurador Hardware (veja exemplo na Figura 5.12, do Capítulo 5). A placa analógica não influencia o endereçamento da placa digital e vice-versa.

A unidade de ampliação digital reserva sempre um espaço para adicionar de 8 bit (bytes) em 8 bit (bytes).

No registro da imagem de processo das entradas estão contidos os estados das entradas digitais. O acesso é somente em leitura.

Bit

Exemplo: I0.1, I3.5, I1.1

Byte, Word, Double word

Exemplo: IB0, IW0, ID2

A cada fim do ciclo de scan da CPU são copiados na saída física os valores armazenados no registro das imagens do processo das

saídas. O acesso é tanto em leitura quanto em escrita.

Bit

Exemplo: Q1.1, Q0.7, Q2.0

Byte, Word, Double word

Exemplo: QB0, QW1, QD0

– Áreas de memória da variável (DB)

A memória dos blocos de dados DB (Data block) pode ser utilizada para armazenar os resultados intermediários de operações matemáticas executadas pelo programa ou para armazenar outros dados relativos ao processo.

Bit

Exemplo: DB10.DBX0.6, DB2, DBX0.2

Byte, Word, Double word

Exemplo: DB20.DBB0, DB10.DBW12, DB12.DBD16

Para tornar mais simples o endereçamento da memória DB, o STEP 7 Basic usa o endereçamento simbólico.

– Áreas de memória do Merker (M)

A área do Merker pode ser utilizada como relé auxiliar para armazenar o estado intermediário de operações lógicas ou de outras informações de controle. O acesso ao Merker é tanto em leitura como em escrita.

Bit

Exemplo: M26.7, M2.0, M100.1

Byte, Word, Double word

Exemplo: MB0, MW100, MD20

– Endereçamento das entradas analógicas (IW)

O conversor A/D da placa analógica converte um valor analógico de entrada de uma grandeza física (tensão, corrente) em um valor digital em formato word (16 bits). Tais valores se acessam mediante o identificador de área (IW).

Exemplo: IW 64, IW 66.

– Endereçamento das saídas analógicas (QW)

O conversor D/A da placa analógica converte um valor digital de saída em formato word (16 bits) em uma corrente ou tensão proporcional ao valor digital. Tais valores se acessam mediante identificador de área (QW).

Exemplo: QW 96, QW 98.

6.5 Configuração dos Endereços com STEP 7 Basic

Quando acrescentamos os módulos de entrada e saída I/O na página da configuração Hardware, os endereços de cada unidade são definidos automaticamente. Veja a Figura 6.8.

A Figura 6.8 mostra o nosso sistema de automação composto de uma CPU 1212C e um módulo de entrada e saída SM.

A Figura 6.9 mostra os detalhes dos componentes.

Com referência à Figura 6.9, cada slot selecionado aparece nas entrelinhas abaixo. Temos:

- Os slots de inserir (slot), os tipos de módulos (modules), os códigos de ordenação (Order no), os endereços das entradas I (addresses I), os endereços das saídas Q (address Q), os comentários (comments), release do componente (firmware), o tipo de componente (type).

Os endereços das entradas e saídas são identificados pela cor branca:

- Entrada e saída discretas on board da CPU 1212C: DI8/DO6, ou seja, CPU com oito entradas e seis saídas com endereço: para entradas de I0.0 a I0.7 e para as saídas de Q0.0 até Q0.5.
- Entradas analógicas on board da CPU 1212C: AI2, ou seja, CPU com dois canais analógicos com endereço: IW64 e IW66 (bytes de 64 a 67).
- Saídas analógicas do módulo SM 1232: AO2 × 14bits com dois canais analógicos com endereço: QW96 e QW98 (bytes de 96 a 99).

6.6 A Programação de Blocos

O PLC S7- 1200 permite a programação linear e estruturada. A programação estruturada é usada para a execução de tarefas de média/alta automação.

As CPUs 1200 permitem a programação estruturada, que na linguagem SIMATIC chama-se **programação de blocos**.

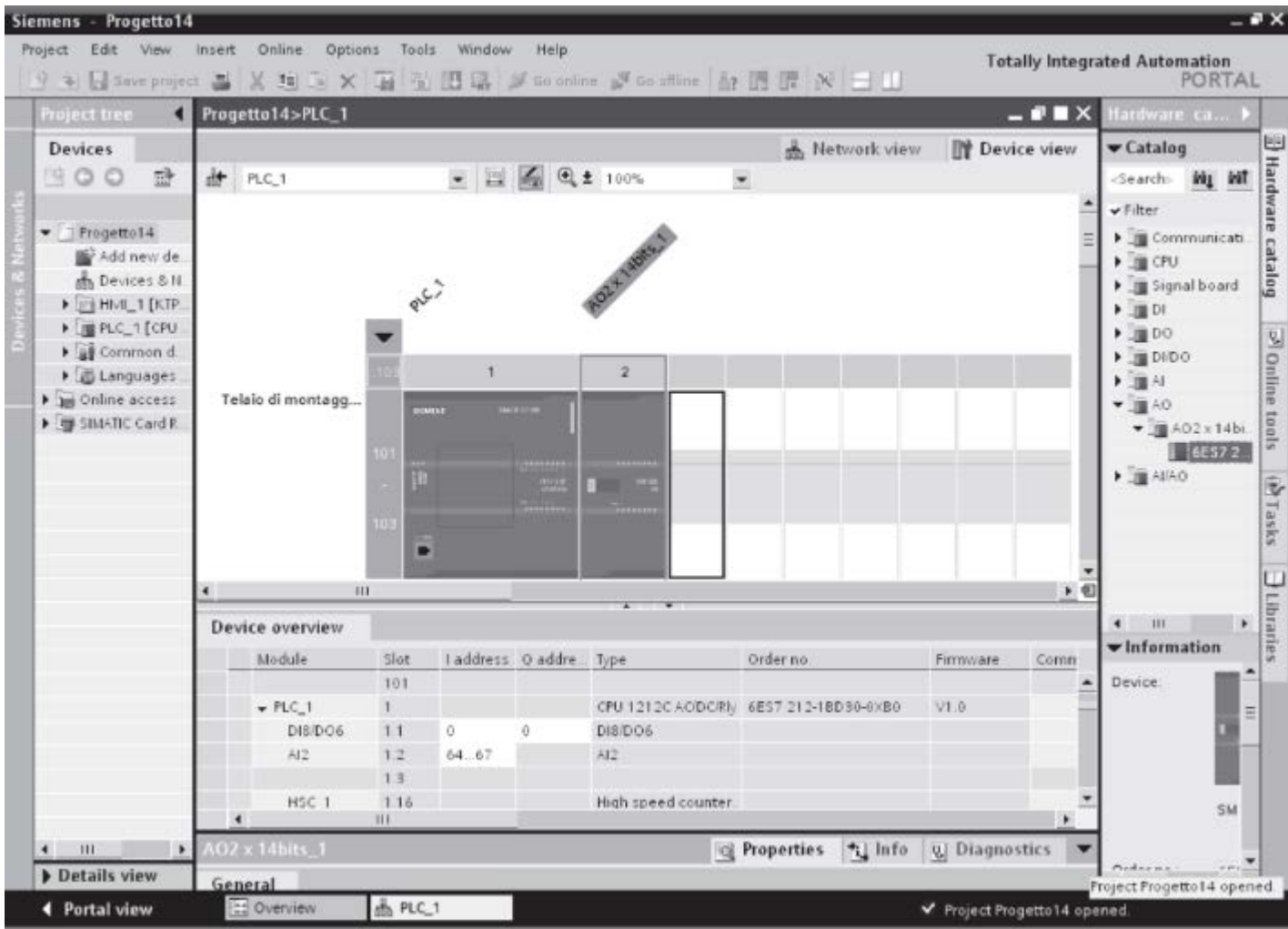


FIGURA 6.8

Device overview								
	Module	Slot	I address	Q address	Type	Order no.	Firmware	Conn
		101						
	▼ PLC_1	1			CPU 1212C AC/DC/Rly	6ES7 212-1BD30-0XB0	V1.0	
	DI8/DO6	1.1	0	0	DI8/DO6			
	AI2	1.2	64...67		AI2			
		1.3						
	HSC_1	1.16			High speed counter..			
	HSC_2	1.17			High speed counter..			
	HSC_3	1.18			High speed counter..			
	HSC_4	1.19			High speed counter..			
	HSC_5	1.20			High speed counter..			
	HSC_6	1.21			High speed counter..			
	Pulse_1	1.32			Pulse generator (PT...			
	Pulse_2	1.33			Pulse generator (PT...			
	► Interfaccia ...	X1			PROFINET interface			
	AO2 x 14bits_1	2		96...99	SM 1232 AO2	6ES7 232-4HB30-0XB0	V1.0	
		3						
		4						
		5						
		6						
		7						
		III						

FIGURA 6.9

6.6.1 Tipos de Blocos

Nos programas no interior do sistema STEP 7 Basic podemos ter os seguintes tipos de blocos:

1. **Organization Block (OB):** os blocos OB executam a estrutura do programa usuário. Os blocos OB são as verdadeiras interfaces entre o sistema operacional da CPU e o programa aplicativo.
2. **Function block (FB):** os FB são blocos com “memória” programável do usuário.
3. **Function (FC):** os FC são blocos do tipo “subroutine”.
4. **Instance Data block (DB):** os Instance Data blocks (DB) são áreas de dados ligadas aos blocos FB. São gerados automaticamente no momento da criação do bloco FB.
5. **Data block(DB):** os DB globais são áreas de dados para o armazenamento dos dados usuários utilizáveis em qualquer ponto do programa.

Em geral nos blocos OB, FC, FB são contidas partes do programa escrito nas linguagens LAD e FBD.

Cada CPU admite um número máximo de blocos OB, FC, FB, DB.

6.6.2 Organização dos Blocos OB

Os blocos OB representam a interface entre o sistema operacional e o programa usuário; contêm a lógica principal do programa.

O OB reage a um evento específico que se verifica na CPU e pode interromper a execução do programa.

O bloco base para a execução do programa é OB1, que determina a estrutura do programa usuário e, portanto, é o único bloco realmente indispensável.

Os outros OB executam funções específicas para a elaboração cíclica dos interrupts, na fase inicial de partida do sistema de automação e para o gerenciamento das falhas.

A elaboração de um OB poderá ser interrompida a qualquer momento, se outro OB quiser ser elaborado. É a *prioridade* que estabelece qual OB pode ser interrompido por outro OB. Obviamente, os OB com prioridade mais elevada podem desligar aqueles com prioridade mais baixa.

Os eventos com a mesma prioridade são elaborados com base na ordem de chegada. Uma vez iniciada, a elaboração de um OB não pode ser interrompida por outro evento do mesmo grupo de prioridade ou de um grupo inferior. Tais eventos são inseridos em uma lista de espera para serem elaborados sucessivamente, permitindo assim à CPU completar a execução do OB atual.

Na Tabela 6.5 são elencados alguns tipos de OB com a relativa prioridade. Outros OB são disponíveis. Nesse caso, a consulta ao manual de sistema é indispensável.

TABELA 6.5

Evento (OB)	Quantidade	Número de OB	Profundidade	Grupo prioridade	Classe prioridade
-------------	------------	--------------	--------------	------------------	-------------------

Ciclo do programa	Um evento de ciclo do programa	1	1	1	1
Start-up	Um evento de start-up	100	1		1
Alarme de atraso	Até quatro eventos de atraso	200 ou maior	8		3
De ciclo	Até quatro intervalos de tempos regulável	200 ou maior	8		4

6.7 Programação Linear e Estruturada

O PLC S7-1200 permite efetuar dois tipos de programação:

- Linear
- Estruturada

Na programação *linear* o programa é memorizado em um único bloco e executado da primeira até a última instrução.

Na programação *estruturada* subdivide-se o programa em blocos distintos e hierarquicamente conectados; cada bloco constitui uma parte do processo. Veja a Figura 6.10.

Os blocos podem ser salvos de modo independente e sucessivamente unidos a fim de se obter o programa principal. Compreende-se, portanto, como se pode intervir modificando um bloco sozinho e mantendo inalterado o resto do programa. Na estrutura deve existir, pelo menos, um bloco principal (OB1), que pode ser subdividido em outros tantos blocos secundários (FC1, FB1, FB5, FC3), que executam determinadas funções e vão e retornam mediante um salto condicionado ou incondicionado do bloco principal aos blocos secundários.

Da programação estruturada em geral se obtêm as seguintes vantagens:

- Os programas de grande dimensão podem ser programados de modo claro;
- Uma parte particular e mais utilizada do programa pode ser padronizada;
- A organização do programa é simplificada;
- A modificação do programa é mais facilitada;
- O teste do programa é simplificado porque pode ser executado por seção.

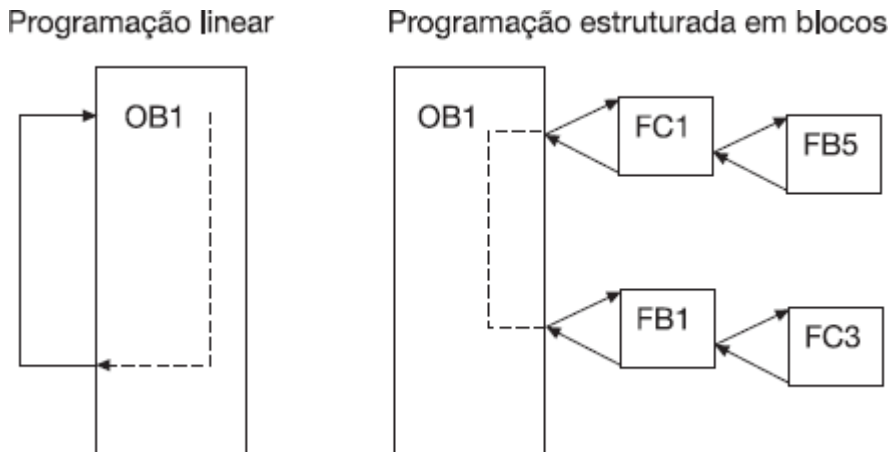


FIGURA 6.10

6.8 Exemplo de um DB Instance e DB Global

Lembramos que os DB global são áreas de dados para o armazenamento dos dados usuários utilizáveis em qualquer ponto do programa.

Em geral nos blocos OB, FC, FB estão contidas partes de programa escritas nas linguagens Ladder e FBD.

Na Figura 6.11 vemos como os *DB global* tipo DB10 podem fornecer os dados a qualquer outro bloco, por exemplo, FC2, FC10, FB2; ou seja, cada bloco pode acessar os dados contidos no DB global DB10.

Já o *DB Instance* DB2 pode fornecer os dados somente ao bloco FB2, ou seja, o acesso ao DB2 é reservado somente ao bloco FB2.

Obviamente no ato da criação do bloco DB temos que escolher se o bloco deve ser do tipo global ou *Instance*.

Na Figura 6.12 temos um exemplo geral de como podem ser usados dois blocos DB Instance com o bloco FB1.

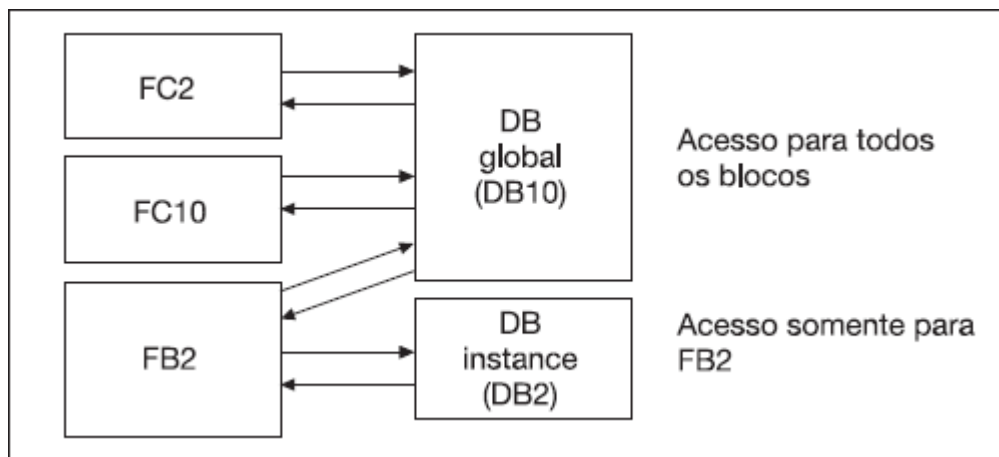


FIGURA 6.11

A programação parte inicialmente do bloco OB1, que torna a chamar o bloco FB1, chamado “Robot”, com a respectiva programação escrita em Ladder ou FBD.

O bloco FB1 torna a chamar constantemente os blocos DB1 e DB2 para ter acesso aos dados relativos:

1. Aos tempos de cada fase de trabalho do robot;
2. Aos dados dos estados de cada fase de trabalho do robot.

A modalidade de trabalho em blocos, como indicado na Figura 6.12, simplifica muito a programação, sobretudo quando é preciso resolver tarefas muito pesadas.

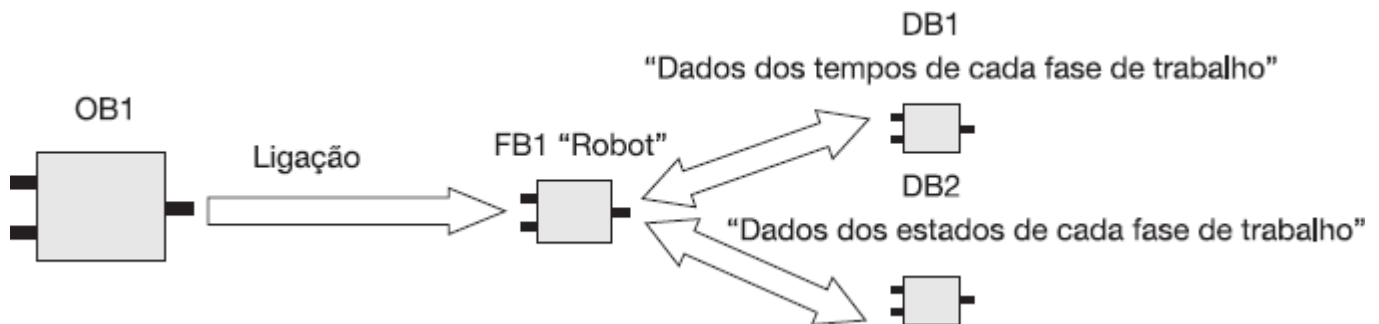


FIGURA 6.12

De fato, subdividir a nossa tarefa chamada “Robot” com as variáveis DB1 e DB2 armazenadas em blocos diferentes apresenta muitas vantagens. A principal é que é possível variar o comportamento do “Robot” sem interferir nas entrelinhas do programa escrito em Ladder ou FBD do bloco FB1, bastando simplesmente variar os parâmetros dos blocos DB1 e DB2.

Com essa modalidade, qualquer operador sem nenhuma experiência de programação em controladores programáveis pode modificar o comportamento do nosso robot em resposta a uma exigência específica do ciclo.

6.9 Blocos com Parâmetros

Os blocos com parâmetros são unidades de programação reutilizáveis previstas pela norma IEC 61131-3.

O corpo do bloco com parâmetros pode ser escrito em Ladder ou FBD.

6.9.1 Exemplo de Bloco com Parâmetros

Na Figura 6.13 temos um exemplo de bloco com parâmetros chamado FC1 dotado de entradas S1, S2, FR e saída KM.

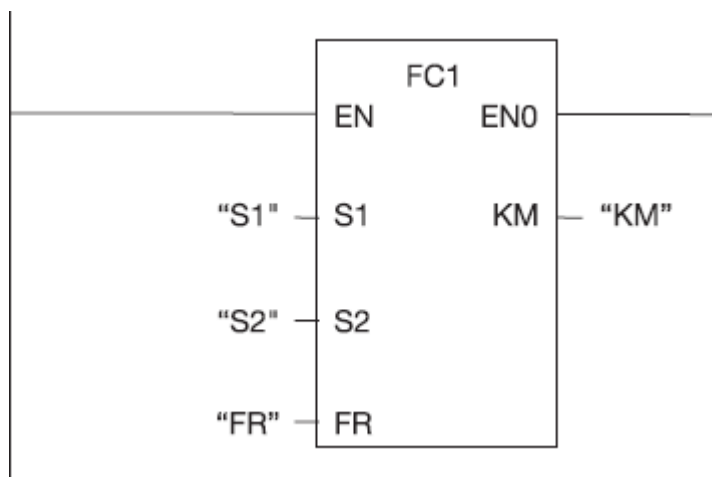


FIGURA 6.13

No interior do bloco temos uma programação escrita, nesse caso, com a linguagem Ladder, conforme a Figura 6.14.

O diagrama Ladder da Figura 6.14 representa uma simples chave de partida para motor trifásico. Notamos como na linguagem SIMATIC os parâmetros de entrada e saída do bloco são indicados como variáveis, chamadas normalmente de “variáveis locais”. Essas variáveis são identificadas com o símbolo “#” e têm validade somente no interior do bloco, no nosso caso FC1 (veja a Figura 6.15).

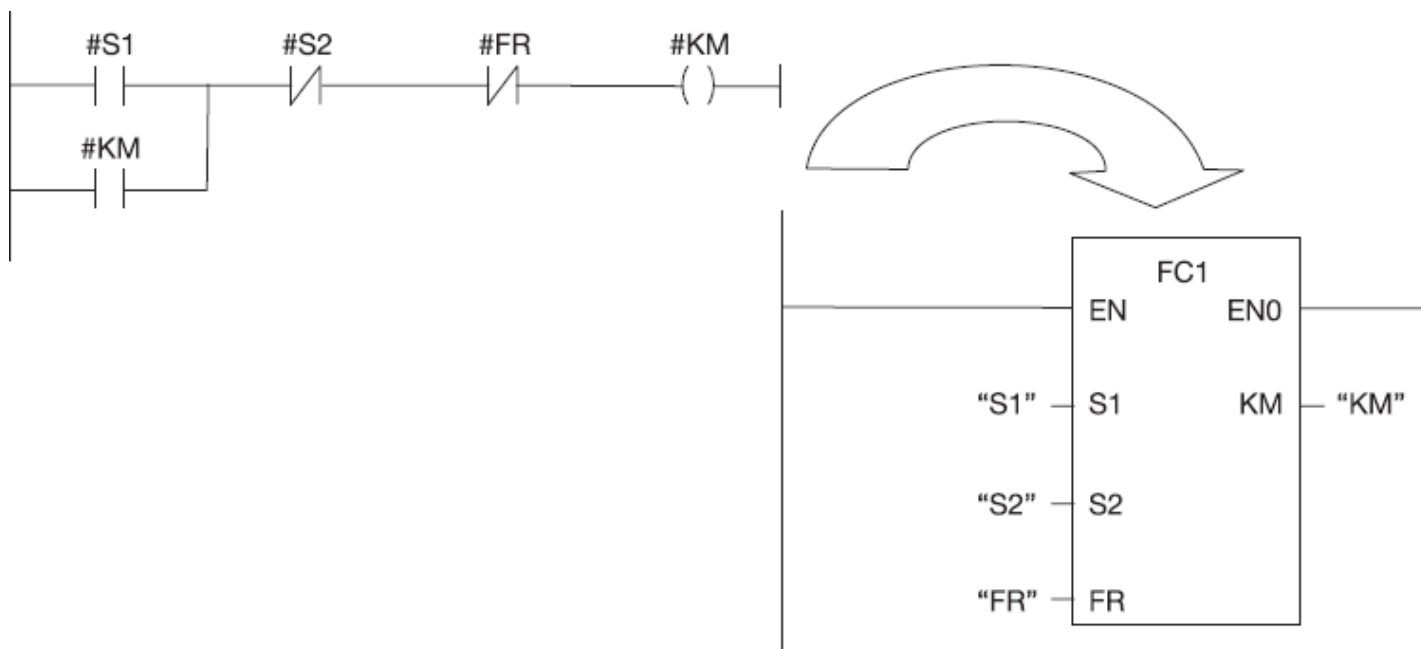


FIGURA 6.14

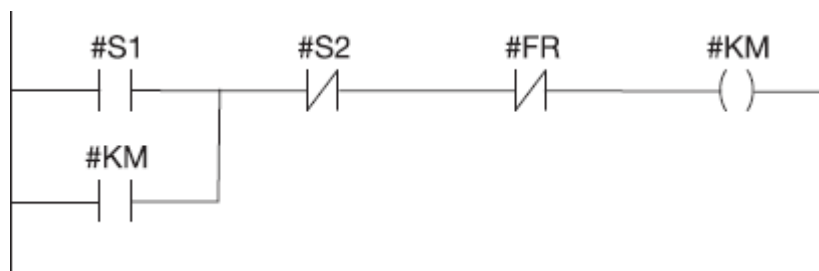


FIGURA 6.15

6.9.2 Configuração de um Bloco com Parâmetros

Em relação ao exemplo de bloco com parâmetros da Seção 6.9.1, se executa agora a configuração desse bloco com STEP 7 Basic.

1. Para que as entradas e saídas do bloco sejam identificadas com o endereçamento simbólico, é preciso ir à tabela dos símbolos e preenchê-la conforme a Figura 6.16.

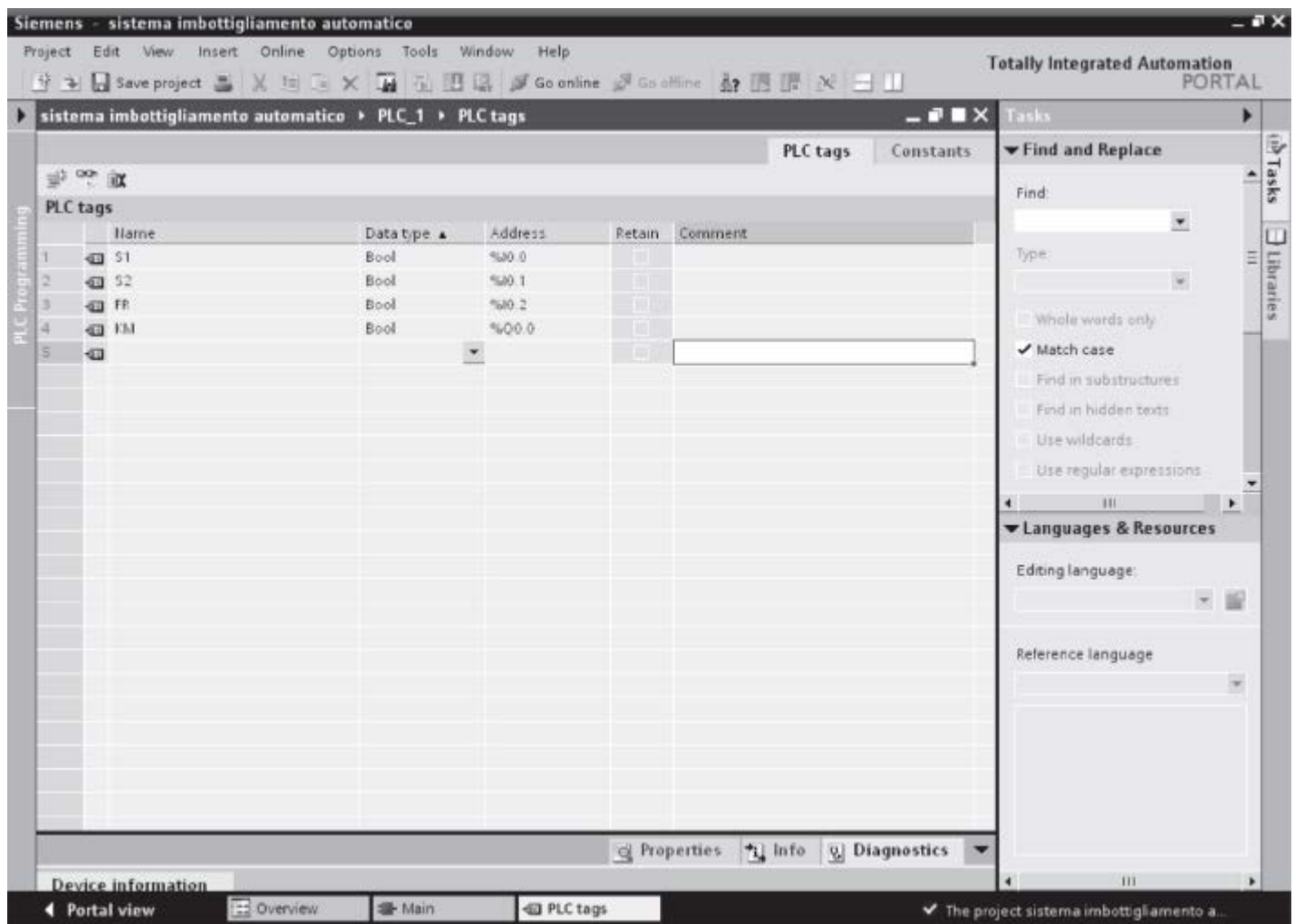


FIGURA 6.16

2. Criar o nosso bloco FC1, conforme a Figura 6.17; confirmar com “OK”.

No interior do bloco, preencher a tabela de declaração das variáveis de entrada *Input* e saída *In/Out* conforme a Figura 6.18, e programar o nosso diagrama Ladder.

Tomando o exemplo relativo a uma chave de partida para motor trifásico, temos como *entrada* (Input):

- S1 botão de start motor
- S2 botão de stop motor
- FR térmica motor

Como saída *In/Out*:

- KM contator

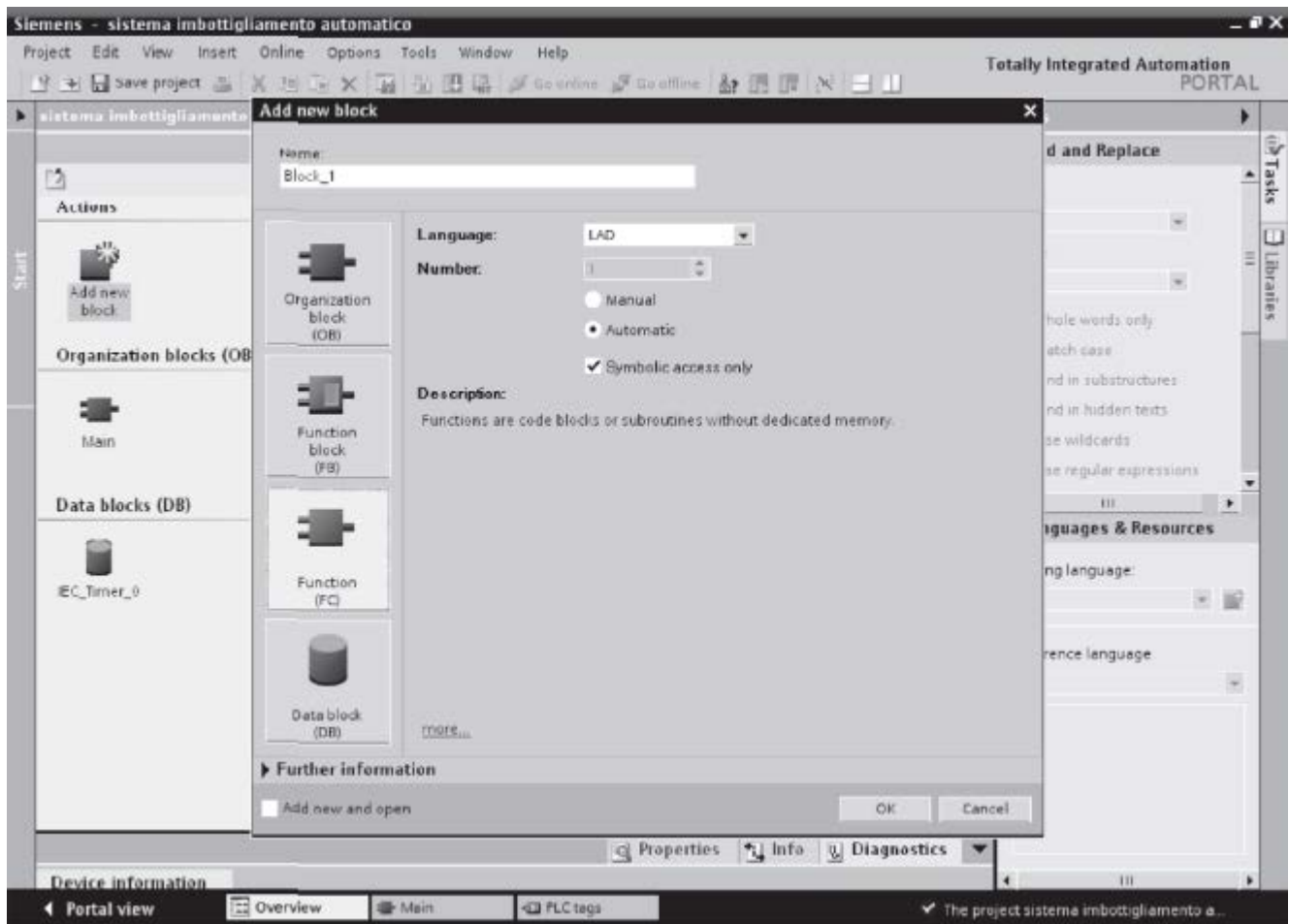


FIGURA 6.17

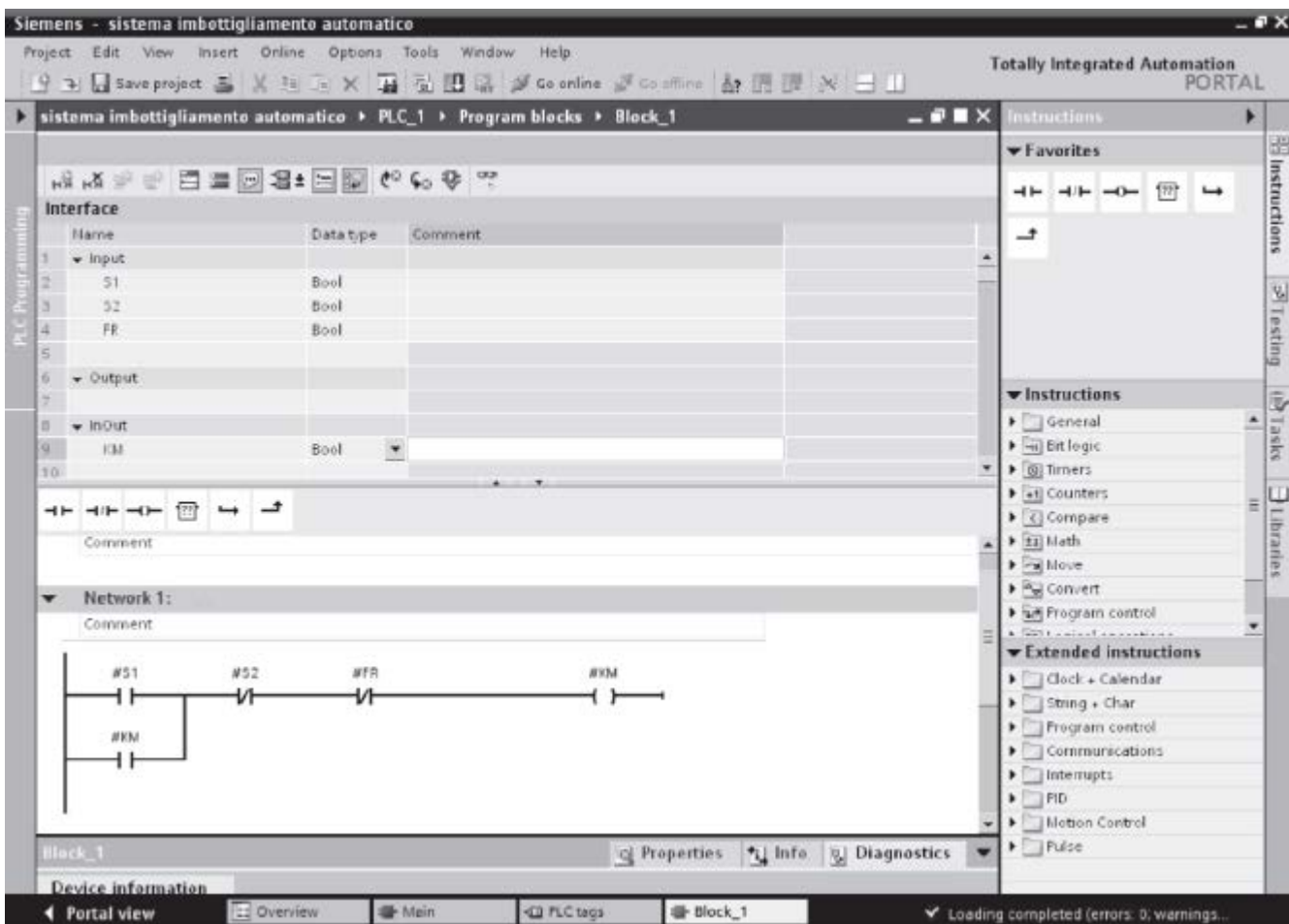


FIGURA 6.18

3. Sair do bloco FC1 que está sendo salvo e entrar na janela da Figura 6.19, clicar sobre o bloco FC1 e copiar (copy) e colar (paste) o

bloco FC1 no bloco principal Main OB1. Logo aparece o nosso bloco parâmetro, conforme Figura 6.20, na linguagem Ladder.

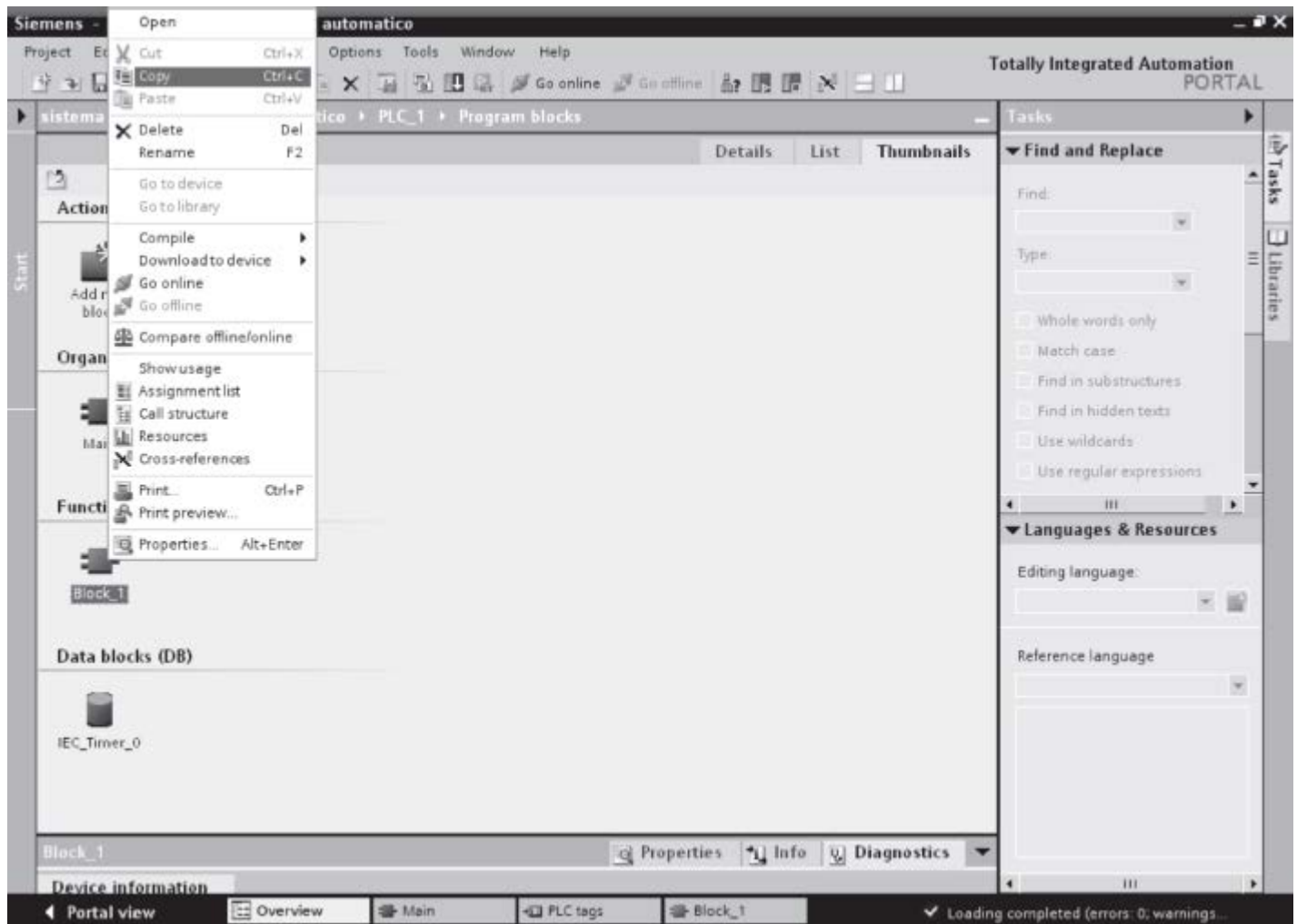


FIGURA 6.19

A simulação do bloco FC1 apresenta-se como na Figura 6.21.

As entradas energizadas estão na condição lógica “1”, enquanto as desenergizadas estão na condição lógica de “0”.

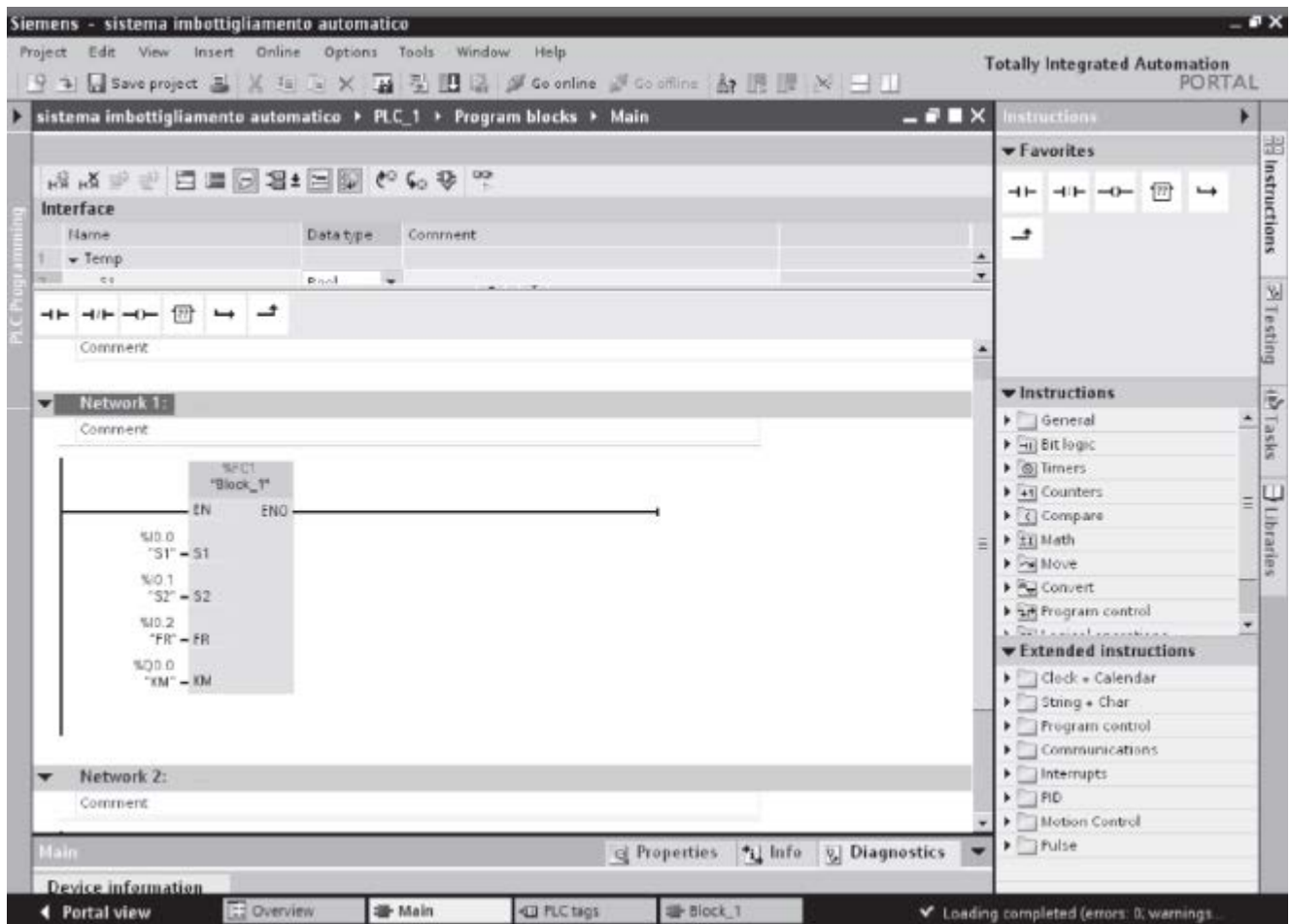


FIGURA 6.20

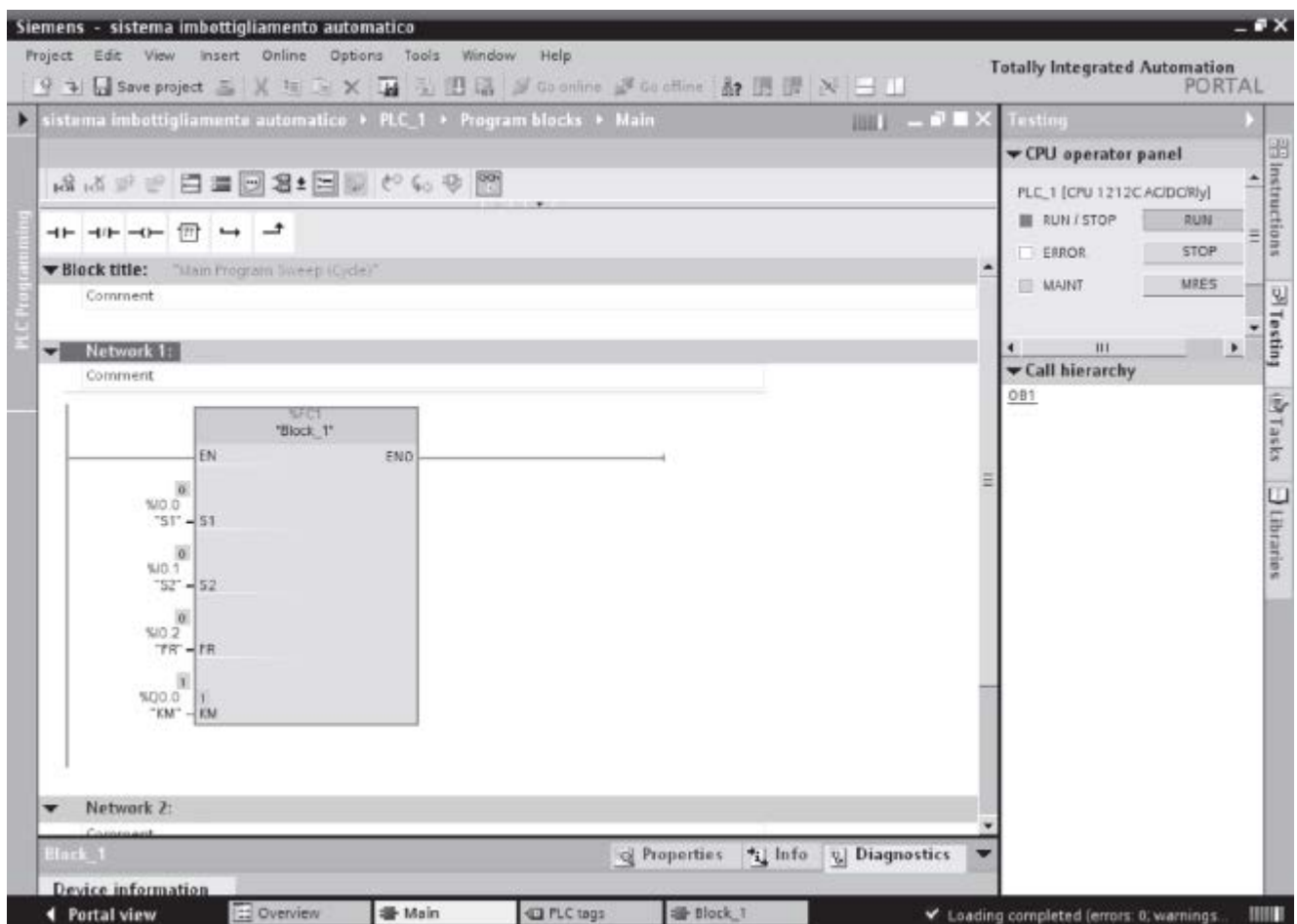


FIGURA 6.21

INTRODUÇÃO ÀS LINGUAGENS DE PROGRAMAÇÃO LADDER E FBD

INTRODUÇÃO ÀS LINGUAGENS DE PROGRAMAÇÃO LADDER E FBD

- 7.0 Generalidades
 - 7.1 O Esquema Funcional Europeu
 - 7.2 O Esquema Funcional Norte-americano
 - 7.3 A Lógica por Contatos
 - 7.4 Símbolos Simples
 - 7.5 Ligação Série e Paralelo
 - 7.6 A Lógica por Portas Digitais
 - 7.7 Transformação de Esquema Funcional em Diagrama Ladder
 - 7.8 Introdução às Aplicações
 - 7.9 A Função SET/RESET
 - 7.10 Aplicação: Relé de Passo
-

7.0 Generalidades

Vimos que a linguagem de programação mais difundida para os controladores é a linguagem com esquema Ladder. O esquema da linguagem LADDER é baseado no esquema a relé. É, portanto, indispensável, para prosseguir neste curso, enfrentar o capítulo relativo à lógica eletromecânica, chamada também de *lógica cabeada* (*wired logic control*).

É importante enfatizar que os símbolos gráficos utilizados pelos esquemas elétricos são baseados na norma internacional IEC 60617-1.13, relativa aos “símbolos gráficos para esquemas elétricos”.

7.1 O Esquema Funcional Europeu

O termo mais similar ao termo esquema funcional se encontra na norma europeia EN 61082-2/5 e é copiado do inglês *Circuit Diagram*, que indica um desenho que transcreve um esquema tanto de comando como de potência de um quadro elétrico a relé.

INTRODUÇÃO ÀS LINGUAGENS DE PROGRAMAÇÃO LADDER E FBD

- 7.0 Generalidades
- 7.1 O Esquema Funcional Europeu
- 7.2 O Esquema Funcional Norte-americano
- 7.3 A Lógica por Contatos
- 7.4 Símbolos Simples
- 7.5 Ligação Série e Paralelo
- 7.6 A Lógica por Portas Digitais
- 7.7 Transformação de Esquema Funcional em Diagrama Ladder
- 7.8 Introdução às Aplicações
- 7.9 A Função SET/RESET
- 7.10 Aplicação: Relé de Passo

7.0 Generalidades

Vimos que a linguagem de programação mais difundida para os controladores é a linguagem com esquema Ladder. O esquema da linguagem LADDER é baseado no esquema a relé. É, portanto, indispensável, para prosseguir neste curso, enfrentar o capítulo relativo à lógica eletromecânica, chamada também de *lógica cabeada* (*wired logic control*).

É importante enfatizar que os símbolos gráficos utilizados pelos esquemas elétricos são baseados na norma internacional IEC 60617-1.13, relativa aos “símbolos gráficos para esquemas elétricos”.

7.1 O Esquema Funcional Europeu

O termo mais similar ao termo esquema funcional se encontra na norma europeia EN 61082-2/5 e é copiado do inglês *Circuit Diagram*, que indica um desenho que transcreve um esquema tanto de comando como de potência de um quadro elétrico a relé.

O termo “esquema funcional” na Europa já está amplamente consolidado e distingue claramente o *esquema de comando* (tipicamente, um circuito com baixa tensão de segurança, 24/48 VAC) do *esquema de potência*, em que é desenhada a carga de potência (normalmente motor trifásico, ou a corrente contínua), circuito com tensão de 230/400 VAC.

No circuito de comando encontram-se todos os botões, relé auxiliar, chaves e outros dispositivos que compõem a parte elétrica de comando do quadro elétrico.

O esquema funcional, em cada caso, é um desenho que se desenvolve com ramificações dispostas em sentido vertical, em cuja parte alta se encontra a fonte de alimentação, denominada normalmente *fase*, e, na parte baixa, o condutor comum, denominado *neutro*. A Figura 7.1 mostra um exemplo de esquema funcional.

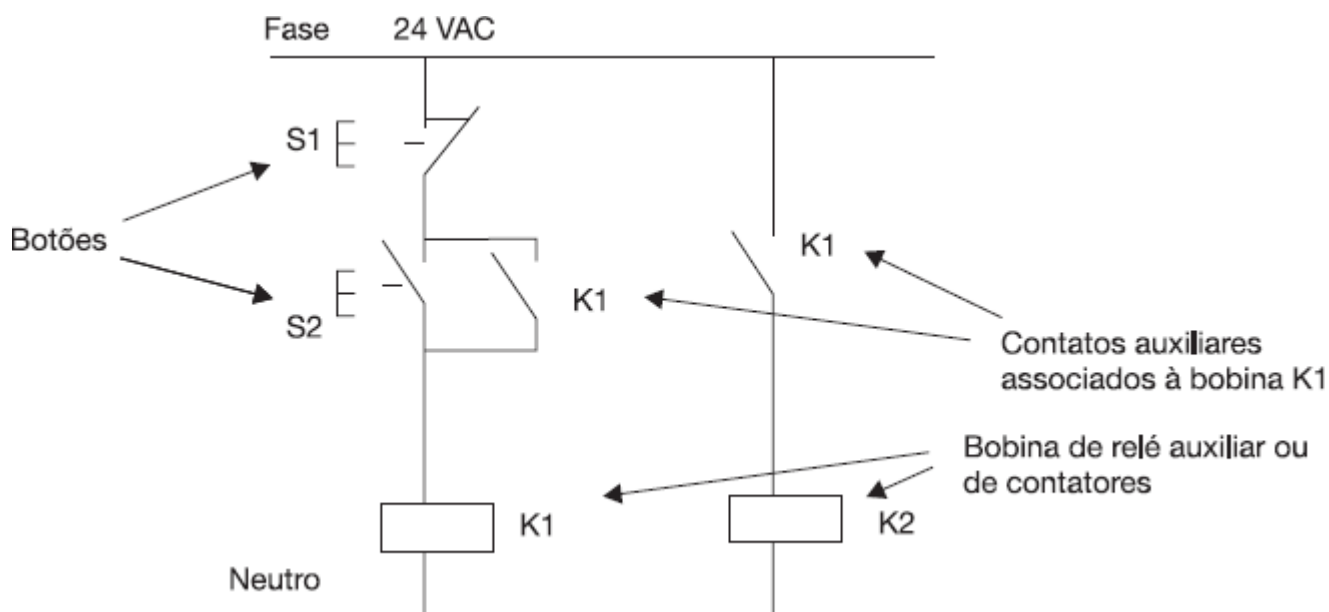


FIGURA 7.1

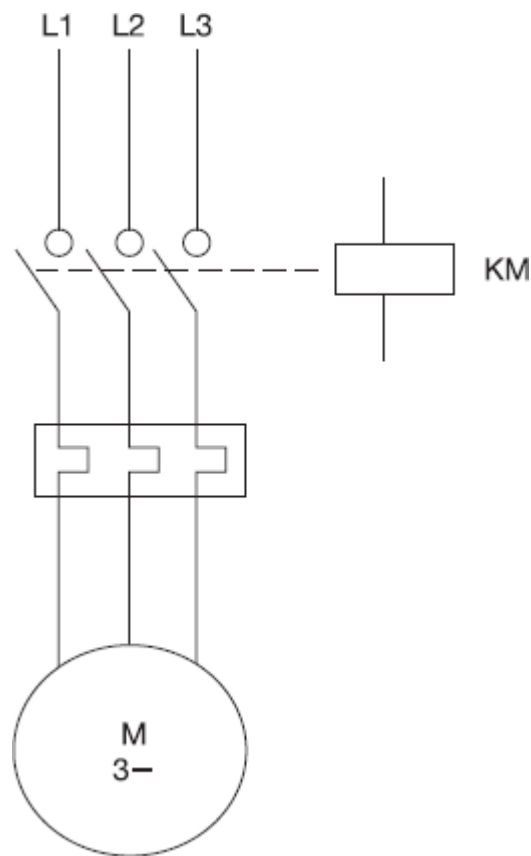


FIGURA 7.2 Exemplo de diagrama de potência (60 Hz, 400 V)

7.2 O Esquema Funcional Norte-americano

Nos Estados Unidos, o esquema de comando de uma máquina é traçado por linhas horizontais e da esquerda para a direita, como na escrita comum. Na prática, é como se rodasse o esquema europeu em 90°. Nesse esquema, partese da esquerda com alimentação e, passando para os vários contatos, se chega à bobina do relé, na parte direita.

Esse modo de fazer os esquemas de comando foi de grande ajuda para a invenção do PLC. Na verdade, o software do PC reproduz no monitor o esquema elétrico. Para um eletrotécnico de formação americana, essa nova tecnologia foi praticamente indolor, ou seja, o técnico recupera a mesma simbologia gráfica do esquema elétrico no monitor do computador. Veja a Figura 7.3.

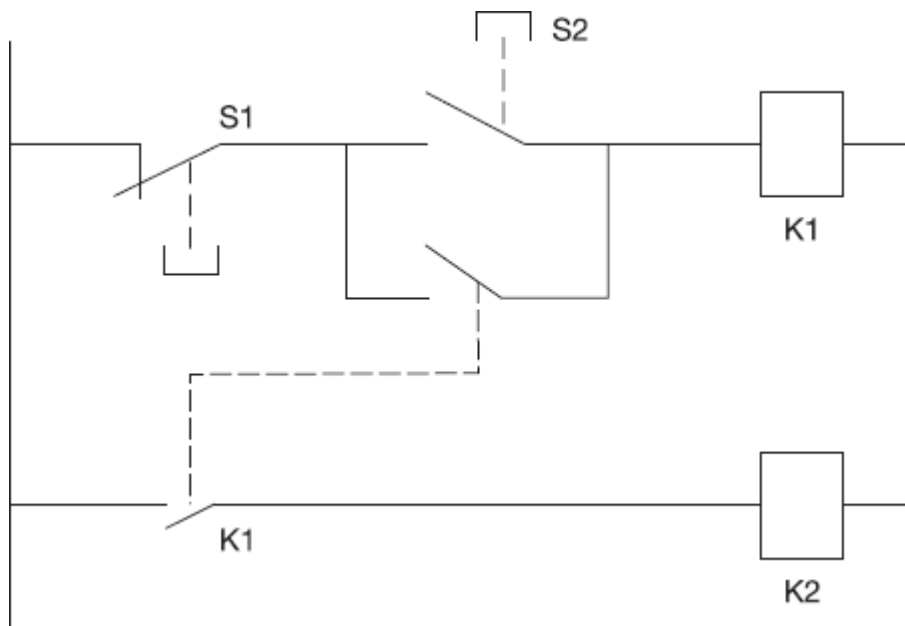


FIGURA 7.3

7.3 A Lógica por Contatos

A seguir será examinada a estrutura de programas por PLC de um diagrama Ladder, chamada também de lógica por contatos.

Vimos como esse tipo de representação apresenta forte analogia com os diagramas elétricos. A lógica de controle digital utiliza somente dois estados, 0 e 1, dos dados. O estado “0” equivale à condição de “falso”, enquanto o estado “1” equivale à condição “verdadeira”.

Nos circuitos elétricos, o estado binário é determinado pelo fluxo de corrente: a presença de corrente equivale a “1”, e sua ausência, a “0”.

Na partida, a cada ciclo de scan a CPU armazena os estados das entradas como 0 (falso) ou 1 (verdadeiro).

– Se a chave é fechada, permite a passagem da corrente.

Nesse caso o estado do circuito é “1”.

“1” = “verdadeiro” = a corrente passa (veja a Figura 7.4).



FIGURA 7.4

– Se a chave é aberta, não permite a passagem da corrente.

Nesse caso o estado do circuito é “0”.

“0” = “falso” = a corrente não passa (veja a Figura 7.5).



FIGURA 7.5

7.4 Símbolos Simples

Na Figura 7.6A é apresentado o contato normalmente aberto de um diagrama elétrico, enquanto na Figura 7.6B temos o equivalente no diagrama por contatos.



(A)



(B)

FIGURA 7.6A E B

Na Figura 7.7A é apresentado o contato normalmente fechado de um diagrama elétrico, enquanto na Figura 7.7B temos o equivalente no diagrama por contatos.



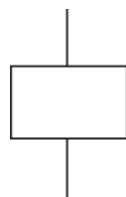
(A)



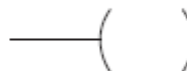
(B)

FIGURA 7.7A E B

Na Figura 7.8 é apresentada a bobina de saída: se o valor “verdadeiro” (corrente ou fluxo de corrente) é passado na bobina, temos a energização da bobina.



(A)



(B)

FIGURA 7.8A E B

Na Figura 7.8A é apresentada a bobina de um diagrama elétrico, enquanto na Figura 7.8B temos o equivalente no diagrama por contatos.

7.5 Ligação Série e Paralelo

Os circuitos elétricos podem ser ligados a operações lógicas.
Por exemplo: uma ligação em série de dois contatos cria uma operação AND de duas entradas.
Veja a Figura 7.9.

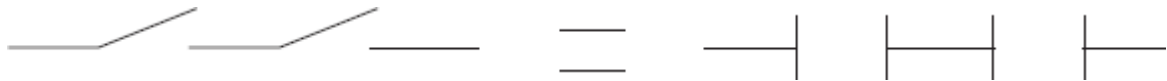


FIGURA 7.9

A corrente elétrica passa somente se os dois contatos estão simultaneamente fechados. Para a lógica OR utiliza-se uma ligação em paralelo de dois contatos. Veja a Figura 7.10.



FIGURA 7.10

A corrente elétrica passa se um dos dois contatos estiver fechado.

7.6 A Lógica por Portas Digitais

O diagrama Ladder (por contatos) pode ser convertido no diagrama equivalente de portas lógicas, normalmente utilizado por técnicos com conhecimento da álgebra booleana. Nesse caso a linguagem chama-se FBD.

Por exemplo, uma ligação em série de dois contatos cria uma operação AND de duas entradas.
Veja a Figura 7.11.

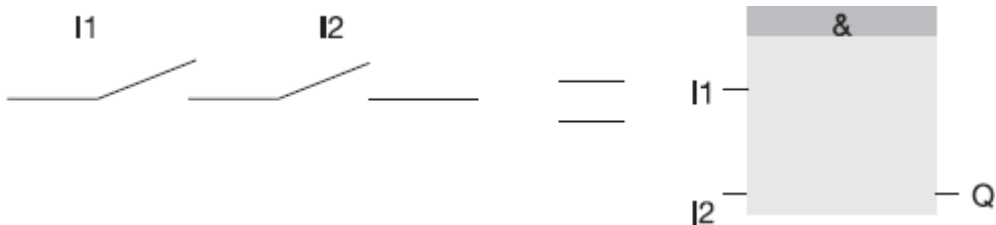


FIGURA 7.11

A ligação em série de dois contatos (um aberto e um fechado) cria uma operação AND de duas entradas com uma porta lógica NOT. A porta lógica NOT identifica um contato normalmente fechado com um pequeno círculo na entrada da porta AND (veja a Figura 7.12).

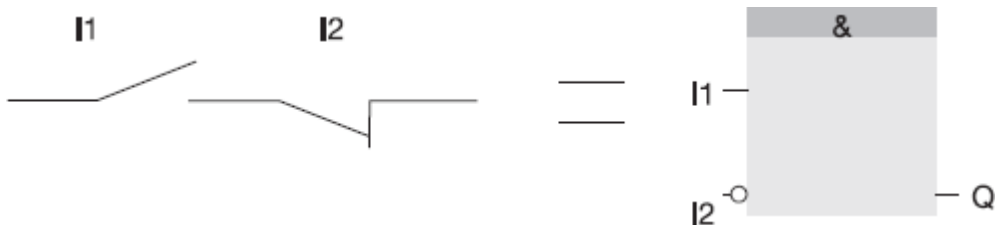


FIGURA 7.12

Assim também uma ligação em paralelo de dois contatos cria uma operação OR de duas entradas (veja a Figura 7.13).

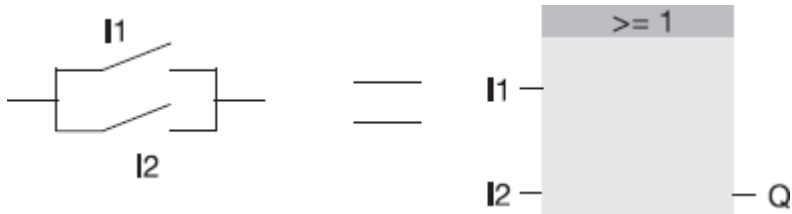


FIGURA 7.13

A bobina de saída da Figura 7.8 apresenta-se como na Figura 7.14.

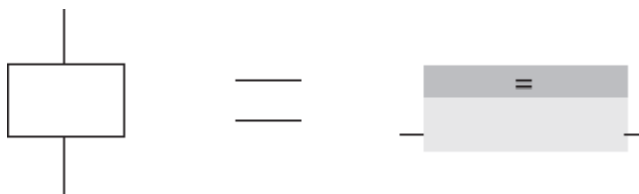


FIGURA 7.14

7.7 Transformação de Esquema Funcional em Diagrama Ladder

O objetivo desta seção é ilustrar regras simples para a transformação de um esquema elétrico funcional em um diagrama Ladder. Existem casos em que a transformação direta do esquema elétrico funcional em diagrama Ladder pode gerar falhas na codificação sucessiva do programa.

Nas figuras a seguir os contatos são indicados com Cx, e as bobinas, com Bx. É importante dizer que na linguagem Ladder cada segmento deve ser finalizado com uma bobina. Antes de ilustrar as várias modalidades de implementação, uma falha grave na programação é aquela que se vê na Figura 7.14A.

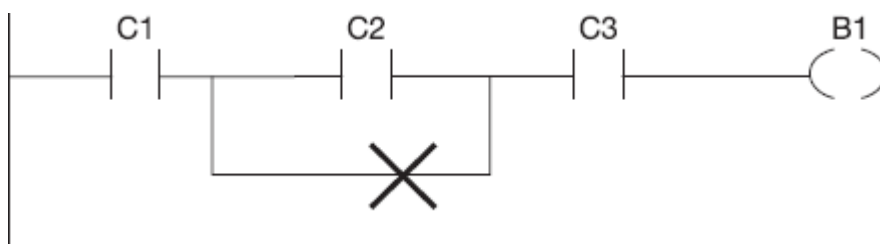


FIGURA 7.14A

Ou seja, não é permitido criar um ramal sem um componente, de forma tal a provocar um curto-circuito do segmento de programação.

– Esquema Elétrico com Ramificação

Se um circuito apresenta uma ramificação como na Figura 7.15A, é obrigatório o diagrama por contatos da Figura 7.15C, e não o da Figura 7.15B.

De fato, não é possível criar um ramal, conforme a Figura 7.15B, que permite uma inversão do fluxo da corrente.

– Esquema Elétrico com Ramificação Complexa

Também nesse caso a tradução do esquema elétrico funcional da Figura 7.16A é correta na Figura 7.16B e incorreta na Figura 7.16C.

– Esquema Elétrico com Pequena Ponte

Na pequena ponte A-B da Figura 7.17A, a sua tradução é correta na Figura 7.17C e incorreta na Figura 7.17B.

Se, por acaso, em vez da pequena ponte A-B for inserido um contato como na Figura 7.18A, o diagrama por contatos tem que ser reescrito como mostra do na Figura 7.18C, porque na Figura 7.18B ele não pode ser codificado diretamente em nenhum modelo de PLC.

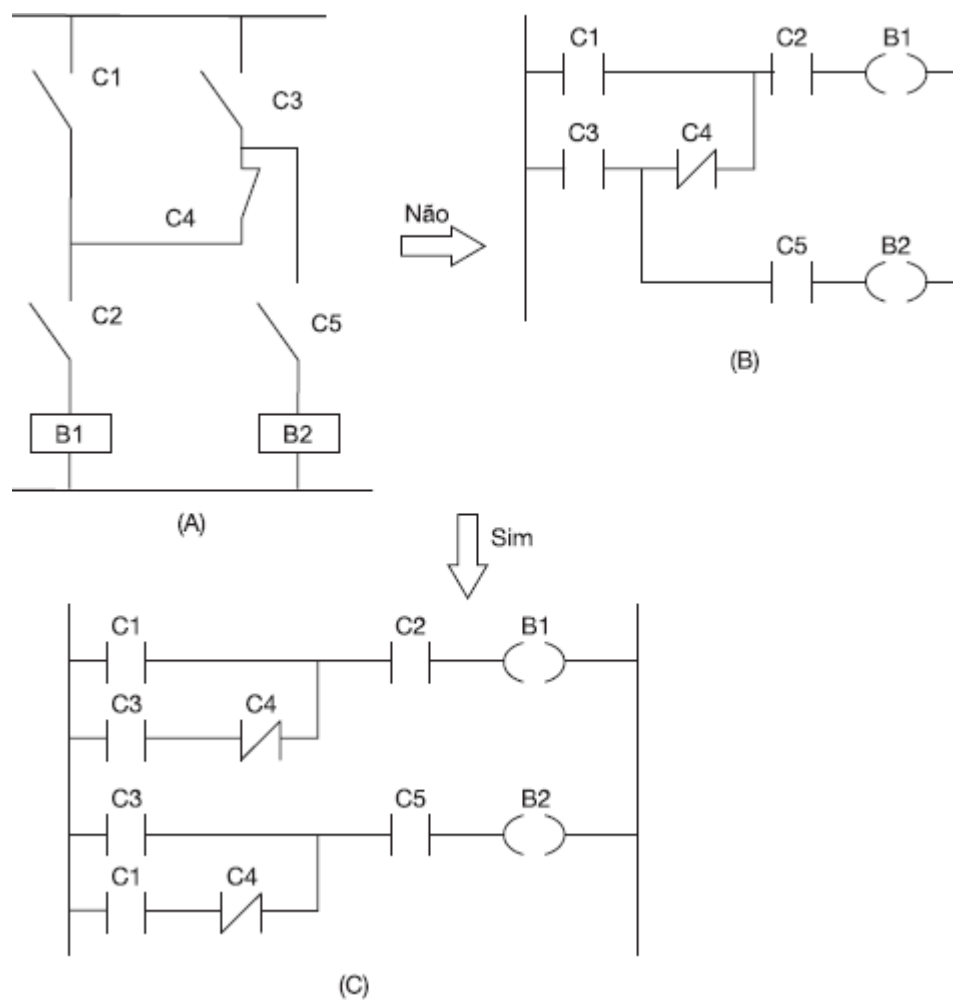


FIGURA 7.15

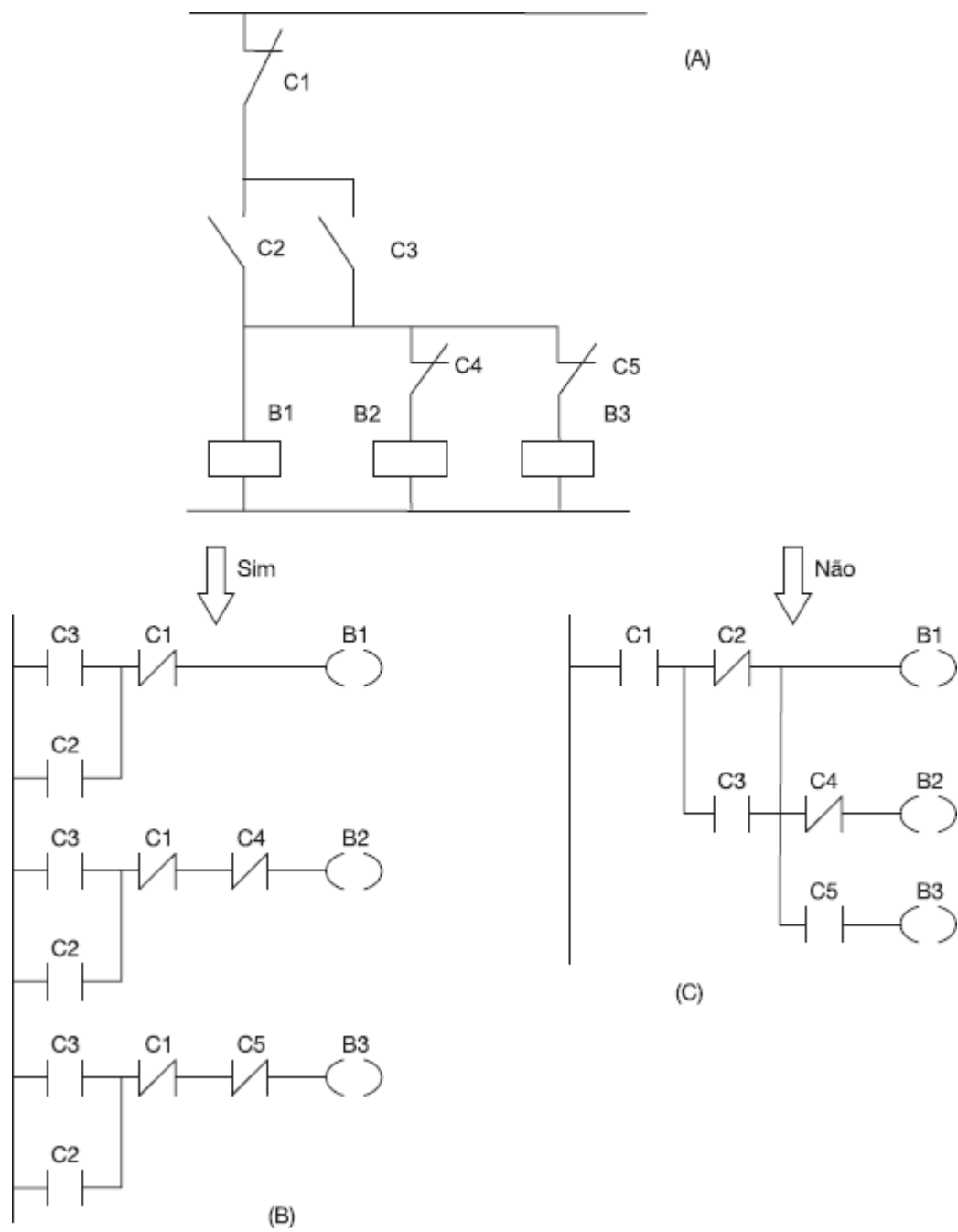


FIGURA 7.16

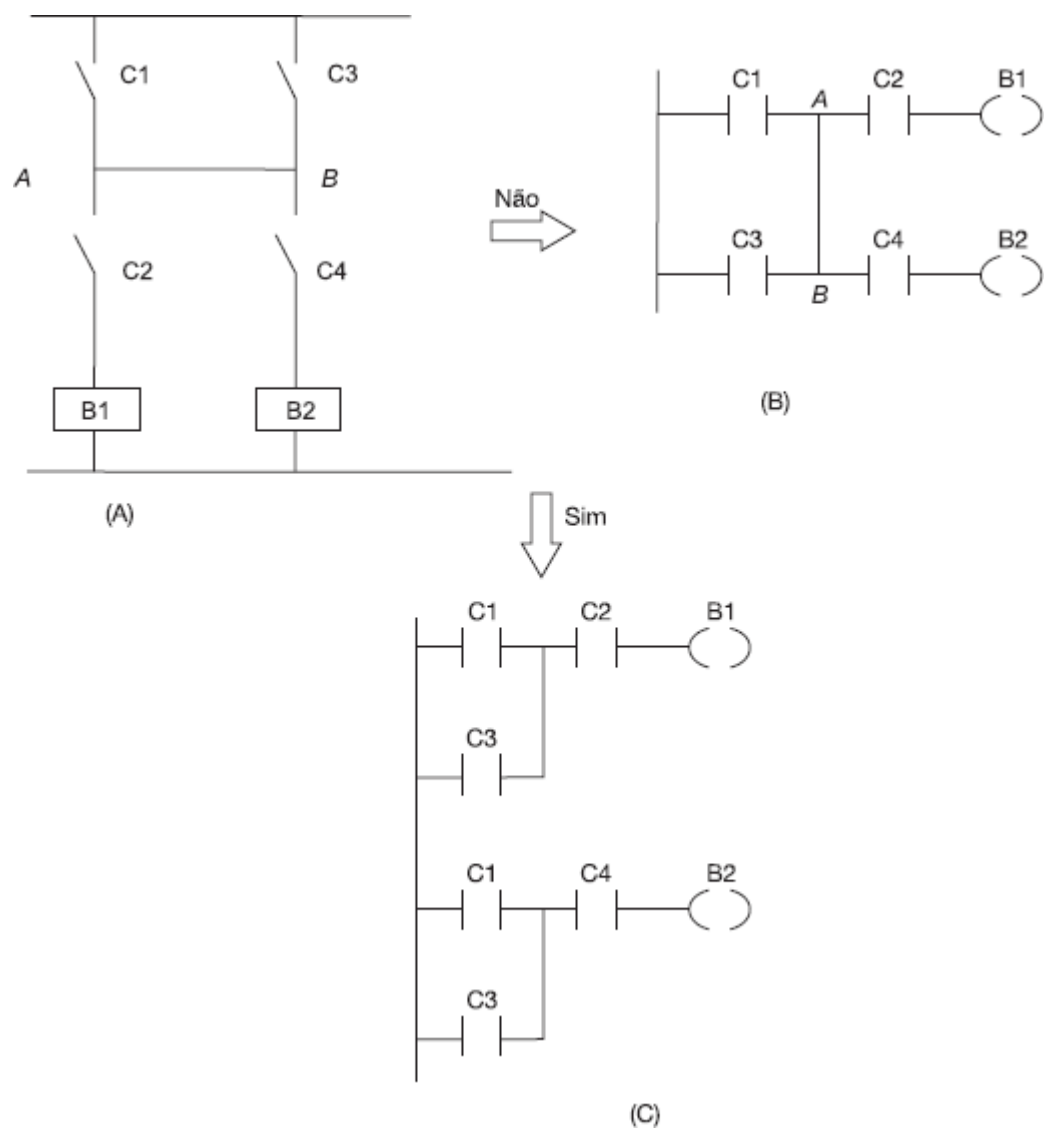


FIGURA 7.17

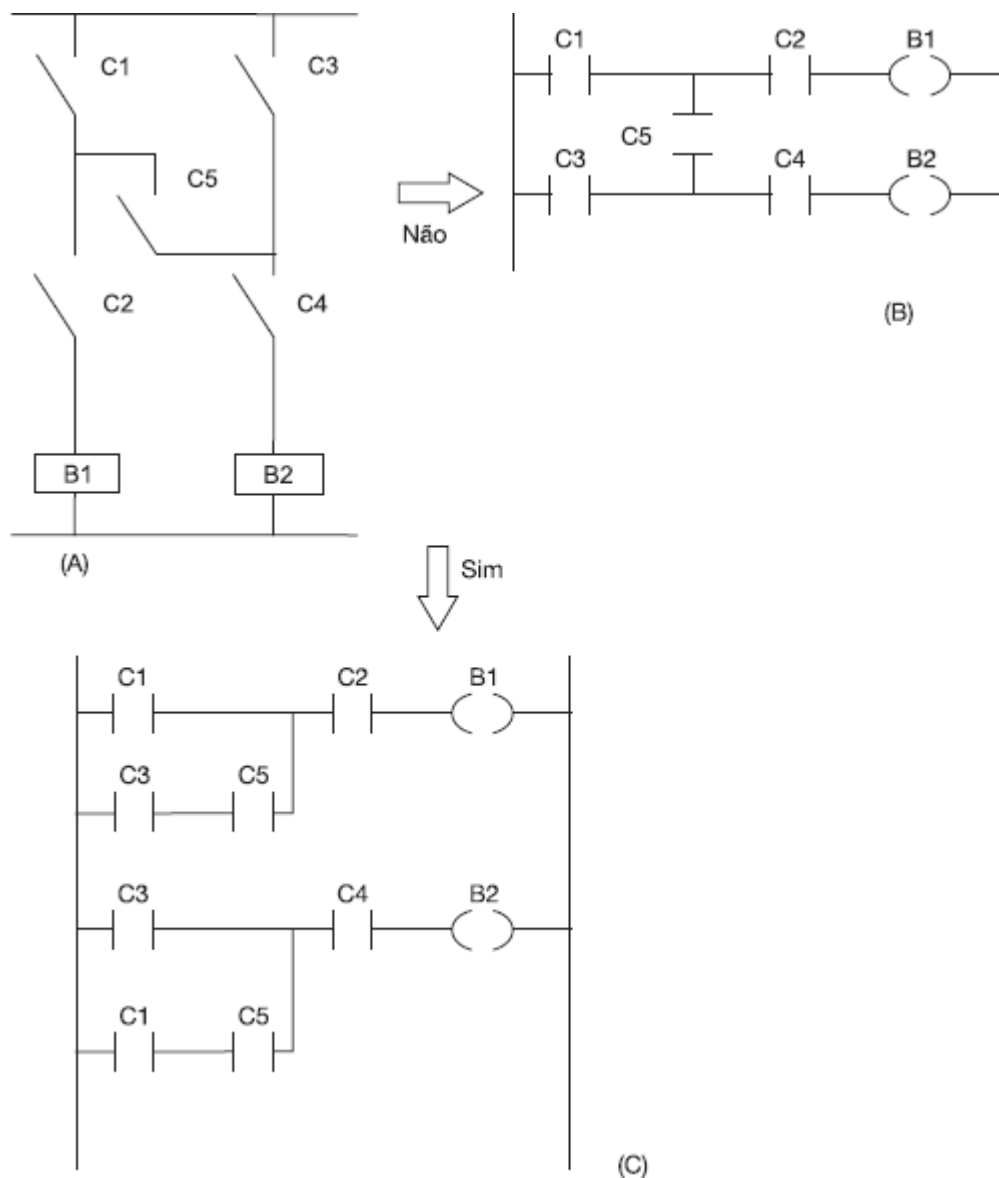


FIGURA 7.18

7.8 Introdução às Aplicações

Nesta seção examinaremos a transformação de esquema elétrico funcional em diagrama Ladder (KOP) de esquemas elétricos simples. Como exemplo, mostraremos esquemas de tipos elétrico e eletropneumático. A intenção dessas aplicações aqui não é examinar o princípio de funcionamento de tal esquema, mas, simplesmente, analisar a conversão do esquema elétrico funcional em diagrama Ladder por ser este compreensível ao PLC.

• Aplicação 1

Transformar o esquema elétrico funcional da Figura 7.19 em relação ao comando eletropneumático de um ciclo automático/semiautomático A+, A– de um cilindro pneumático de duplo efeito com eletroválvula biestável em diagrama Ladder.

Nota: O fim de curso Fca0 no esquema funcional é desenhado normalmente aberto, porém, quando for instalado na realidade, é fechado, porque o cilindro, por ser dentro, toca a chave fim de curso “Fca0”, fechando-se.

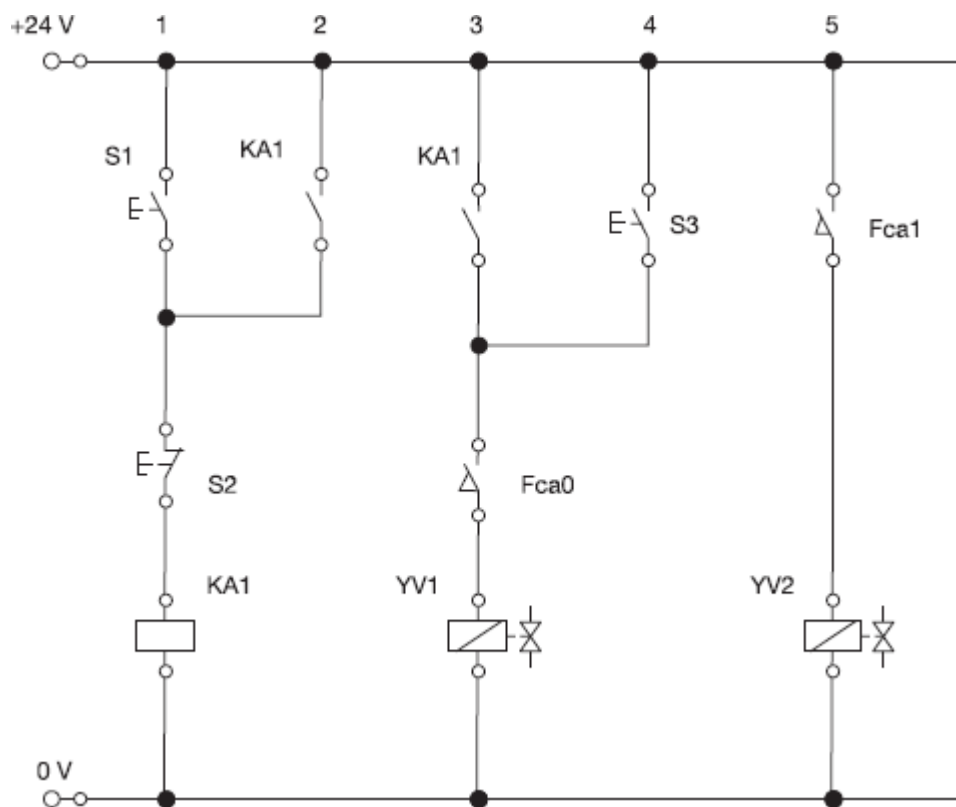


FIGURA 7.19

Diagrama Ladder do esquema funcional referente à Figura 7.19 (veja a Figura 7.20).

Diagrama FBD do esquema funcional referente à Figura 7.20 (veja a Figura 7.21).

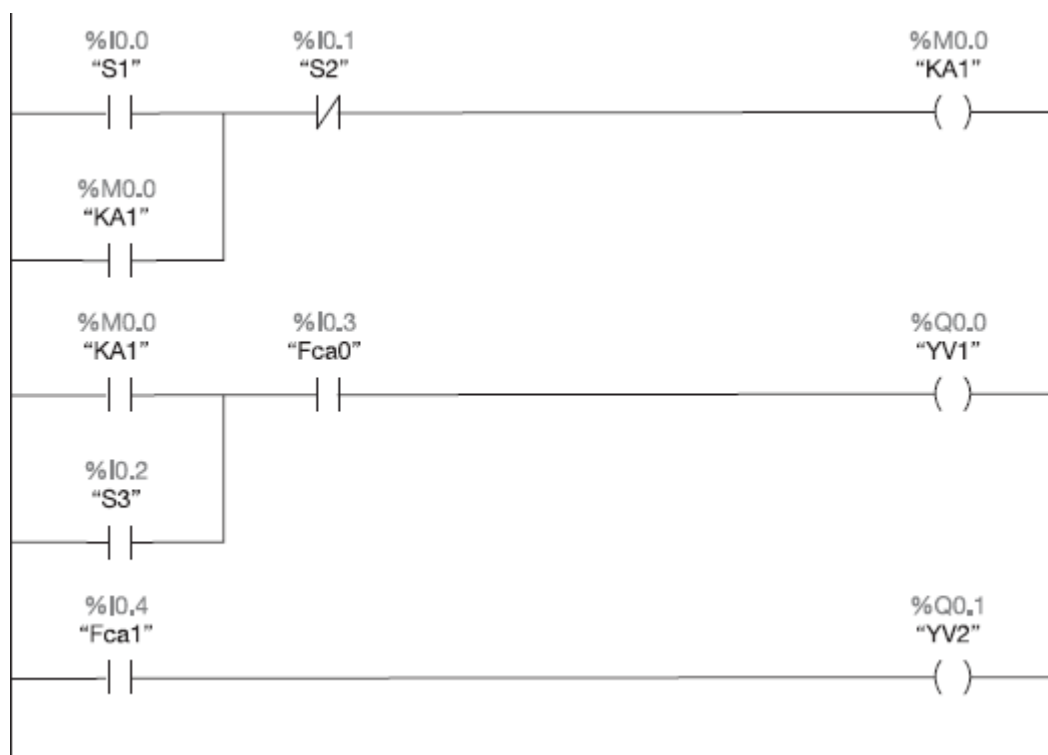


FIGURA 7.20

TABELA 7.1 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão automático
S2	I0.1	Botão de parada
S3	I0.2	Botão semiautomático
Fca0	I0.3	Fim de curso cilindro dentro

Fca1	I0.4	Fim de curso cilindro fora
KA1	M0.0	Merker de relé auxiliar
YV1	Q0.0	Eletroválvula saída cilindro A+
YV2	Q0.1	Eletroválvula recuo cilindro A-

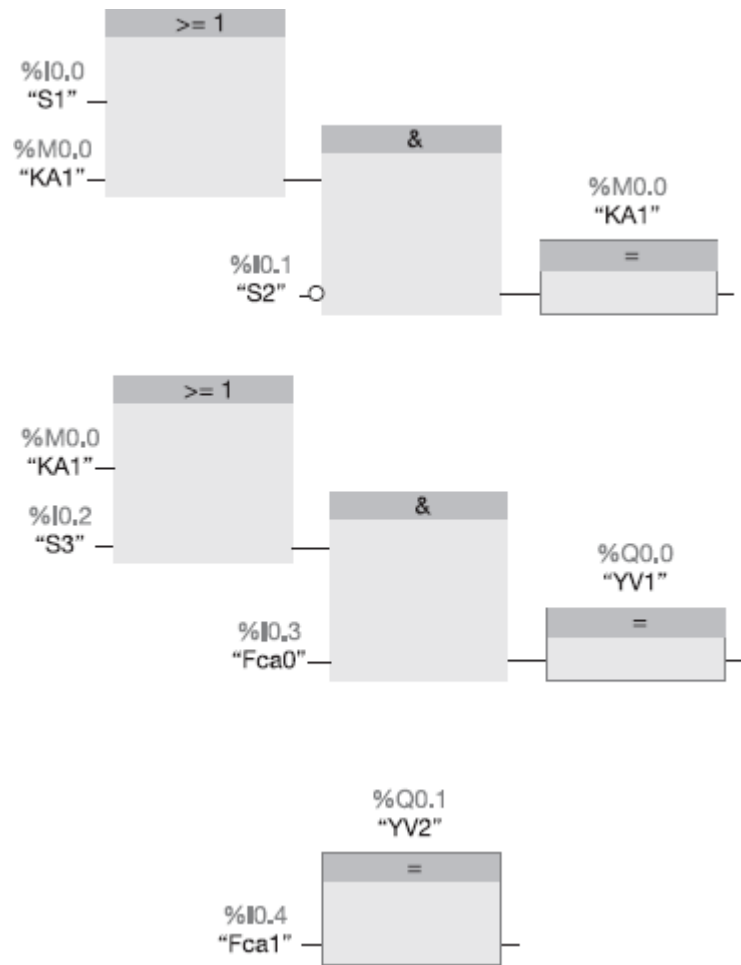


FIGURA 7.21

• Aplicação 2

Circuito de comando eletropneumático do ciclo automático A+, A– de um cilindro de duplo efeito com eletroválvula monoestável. Veja a Figura 7.22.

Nota: O fim de curso Fca0 no esquema funcional é desenhado normalmente aberto, porém, quando for instalado na realidade, é fechado, porque o cilindro, por ser dentro, toca a chave fim de curso “Fca0”, fechando-se.

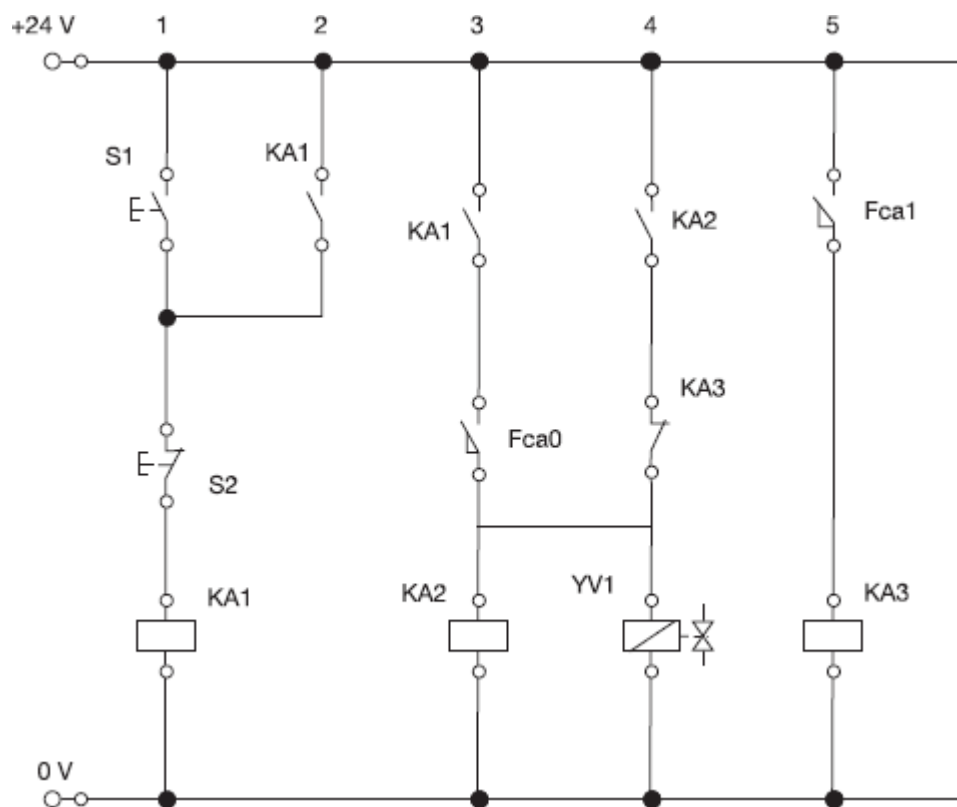


FIGURA 7.22

TABELA 7.2 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão automático
S2	I0.1	Botão de parada
Fca0	I0.3	Fim de curso cilindro dentro
Fca1	I0.4	Fim de curso cilindro fora
KA1	M0.0	Merker de relé auxiliar
YV1	Q0.0	Eletroválvula saída cilindro
KA2	M0.1	Merker de relé auxiliar
KA3	M0.2	Merker de relé auxiliar

Diagrama Ladder do esquema funcional referente à Figura 7.22 (veja a Figura 7.23).

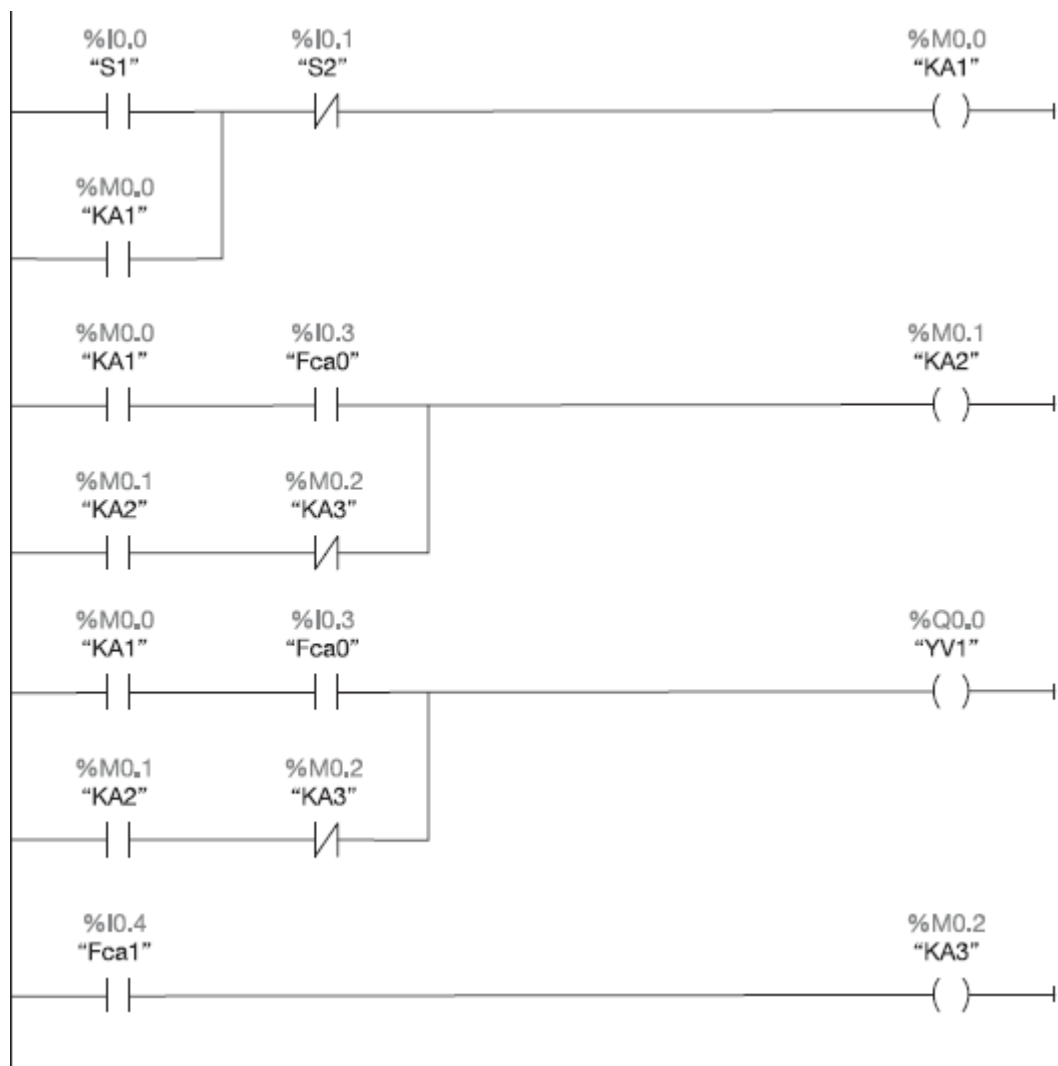


FIGURA 7.23

• Aplicação 3

Inversão do sentido de rotação do MAT sem passagem pelo stop com sinalização. Veja a Figura 7.24.
Diagrama Ladder do esquema funcional referente à Figura 7.24 (veja a Figura 7.25).

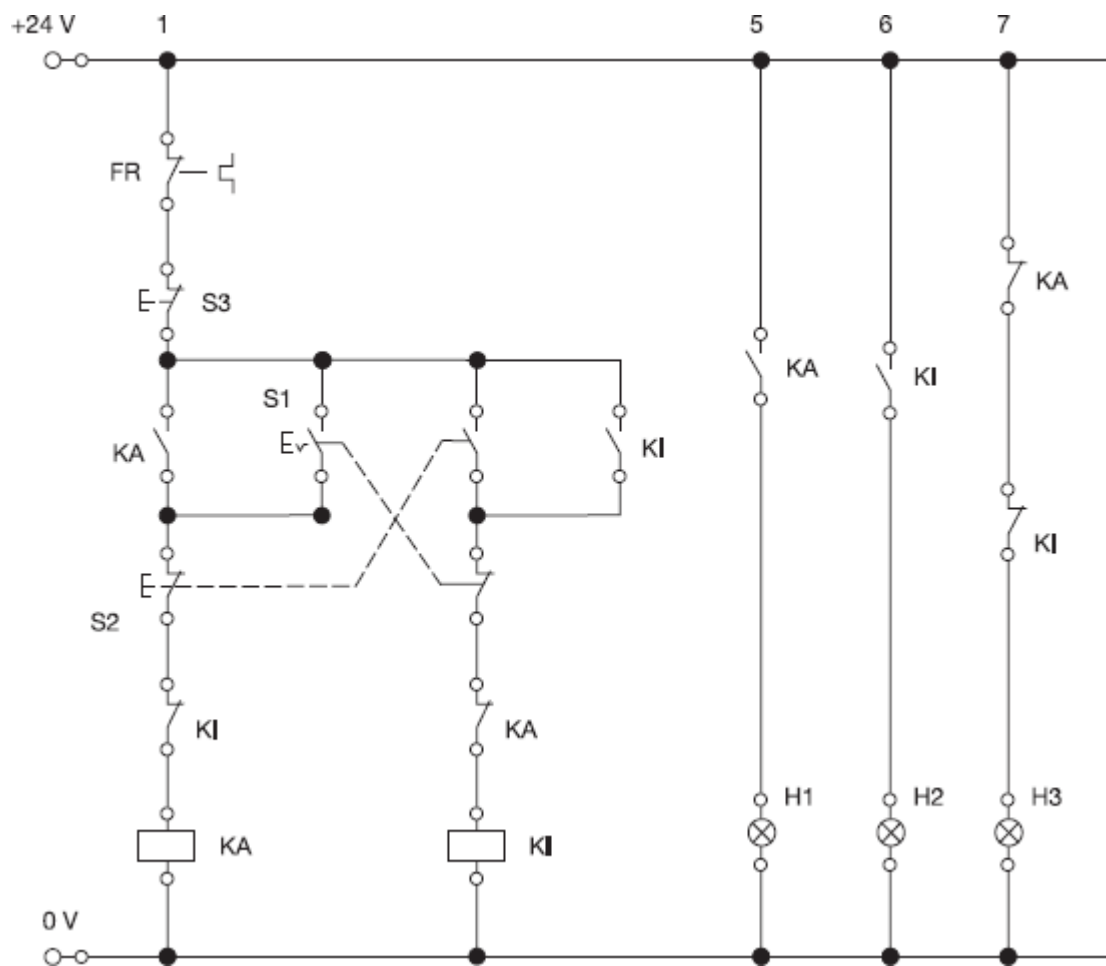


FIGURA 7.24

TABELA 7.3 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão de partida para a frente
S2	I0.1	Botão de partida para trás
S3	I0.2	Botão de parada
FR	I0.3	Térmica
KA	Q0.0	Contator motor para a frente
KI	Q0.1	Contator motor para trás
H1	Q0.2	Lâmpada motor para a frente
H2	Q0.3	Lâmpada motor para trás
H3	Q0.4	Lâmpada de parada motor

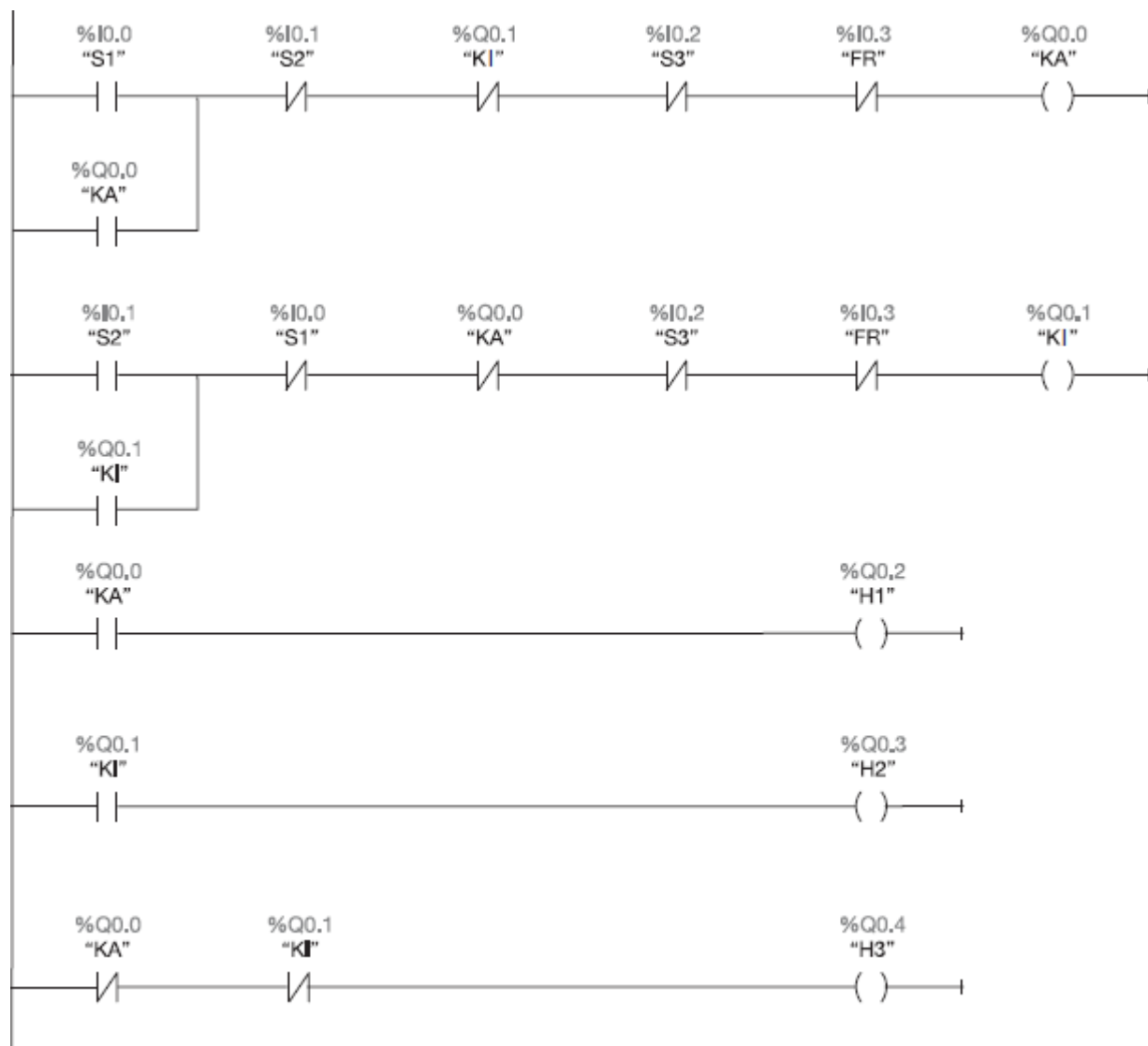


FIGURA 7.25

7.9 A Função SET/RESET

A operação de **set** pode ser vista como imposição de uma saída em autorretenção. Com essa instrução da CPU S7-1200, pode-se impor 1 bit; ela é tradicionalmente utilizada com a instrução **reset**. Os dois trabalham com o mesmo bit. Na Figura 7.26 vê-se um exemplo para ativação de uma saída utilizando as instruções set e reset e dois botões normalmente abertos. Acionando o botão I0.0 ativa-se a saída Q0.0 (set) com autorretenção.

Como representado na Figura 7.26, acionando o botão I0.1 desativa-se Q0.0 (reset). É interessante notar que acionando-se simultaneamente I0.0 e I0.1, Q0.0 permanece desativado. Tecnicamente, diz-se que o **reset é dominante**. Numa situação inversa, se a linha de reset do exemplo da Figura 7.26 antecede a linha de set, diz-se que o **set é dominante**, acionando a saída Q0.0. Essa situação se deve à lógica do ciclo de scan do PLC. De fato, o programa é executado pela CPU de cima para baixo e da esquerda para a direita.

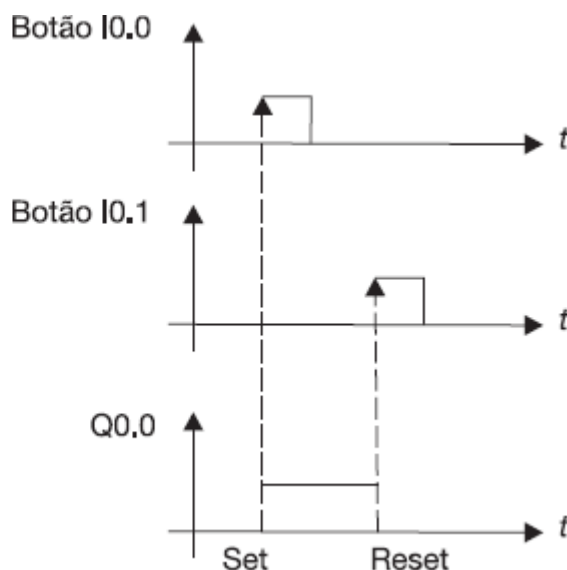
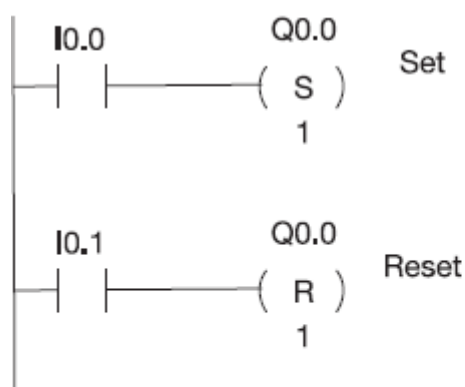


FIGURA 7.26

7.10 Aplicação: Relé de Passo


Se deseja realizar uma simples aplicação denominada relé de passo, o funcionamento é o seguinte:

- pressionando-se o botão de partida S1, energiza-se a saída K1;
- pressionando-se de novo o mesmo botão, desenergiza-se a mesma saída.

TABELA 7.4 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão de partida/parada
K1A	M0.0	Merker de relé auxiliar
K2A	M0.1	Merker de relé auxiliar
K3A	M0.2	Merker de relé auxiliar
K4A	M0.3	Merker de relé auxiliar
K1	Q0.0	Saída

Diagrama Ladder do relé de passo (veja a Figura 7.27).

Nota: O símbolo “” é um contato chamado “transição positiva”, e se fecha por um ciclo de scan, exatamente a cada transição off/on. Será examinado no próximo capítulo.

O funcionamento do esquema Ladder do relé de passo é o seguinte: quando se pressiona o botão S1, por meio do contato a transição positiva se energiza a bobina K1A. Em consequência, fecha-se o contato auxiliar K1A e se energiza a bobina K2A, porque o contato K4A está fechado.

Além disso, também sendo fechado o contato K2A, a bobina K4A é setada, e, em consequência, se ativa a saída K1. Pressionando-se novamente o mesmo botão S1, se encontra dessa vez o contato K4A fechado em série com a bobina K3A; aberto o contato K4A em série com a bobina K2A, energiza-se a bobina K3A. Fechado o contato K3A (enquanto a mesma bobina estiver energizada), se resseta a bobina K4A e, em consequência, se desenergiza a saída K1.

É interessante observar nessa aplicação que a frequência dos pulsos elétricos que chegam à saída K1 é a metade da frequência dos pulsos elétricos da entrada S1. Lembremos que a transição negativa on/off não tem nenhum efeito na saída K1. Portanto, sua característica é dividir por dois o sinal de entrada. A função do relé passo-passo é idêntica àquela que em eletrônica digital se chama *biestável* ou *flip-flop tipo T*.

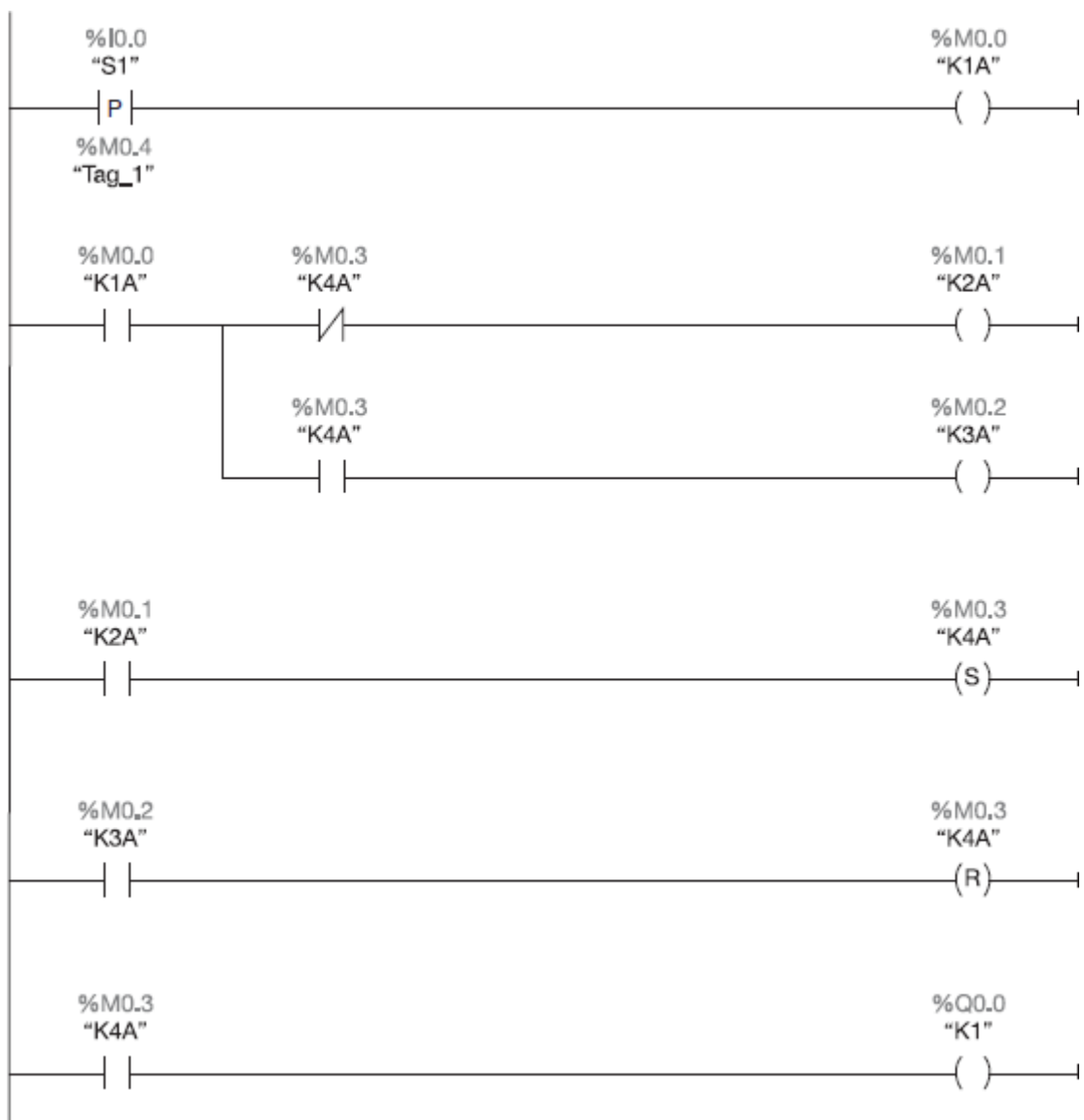


FIGURA 7.27

Diagrama FBD do relé de passo (veja a Figura 7.28).

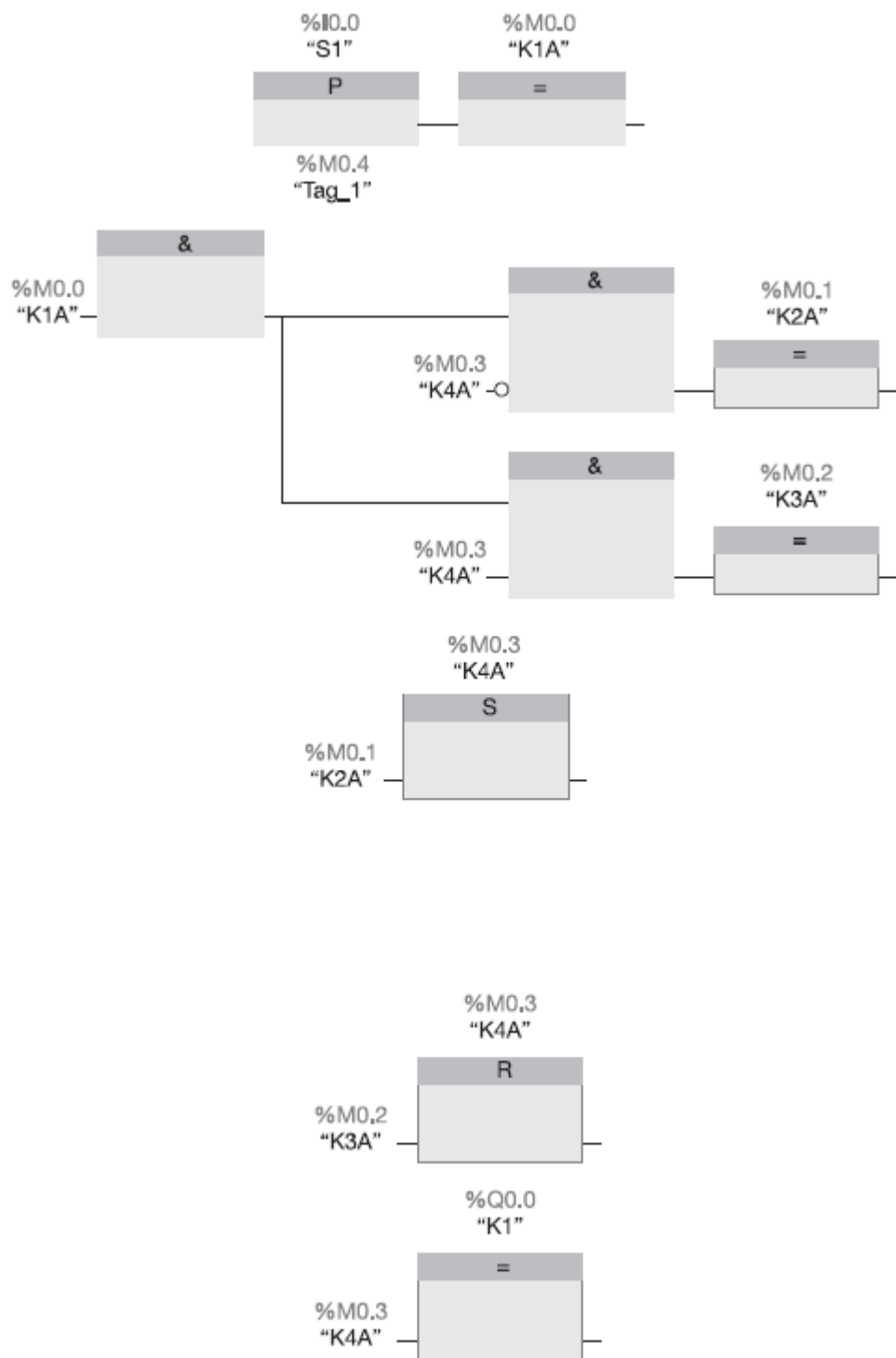


FIGURA 7.28

DETECÇÃO DE BORDA DE DESCIDA E SUBIDA

DETECÇÃO DE BORDA DE DESCIDA E SUBIDA

- 8.0 Generalidades
 - 8.1 Definição de Borda de Descida e Subida
 - 8.2 Construção da Operação de Detecção da Borda de Descida e Subida
 - 8.3 Operação Especial de Detecção com a CPU S7-1200
 - 8.4 Aplicação: Partida e Parada com Segurança de um MAT (Motor Assíncrono Trifásico)
-

8.0 Generalidades

Em automação industrial, frequentemente é necessária a gestão de um sinal no momento em que este muda de estado, e não quando está em um estado definido. Essa gestão de mudança de estado de um sinal pode ser útil na lógica de controle de qualquer máquina industrial.

Ocorre, muitas vezes, que a lógica de controle precise apenas detectar a mudança de estado de um botão de comando, por exemplo, quando, por falha do operador, se pressiona de forma constante o botão de acionamento da máquina.

Esse procedimento pode causar uma movimentação perigosa da máquina. Se ao contrário detectamos a mudança de estado do botão, essa movimentação perigosa da máquina não acontece, tornando a instalação em condição de plena segurança.

8.1 Definição de Borda de Descida e Subida

Referimo-nos à borda de subida de um sinal elétrico quando este passa de um estado a “0” lógico a um estado a “1” lógico ou de off a on. Referimo-nos à borda de descida de um sinal elétrico quando este passa de um estado a “1” lógico a um estado a “0” lógico ou de on a off.

A função de detecção da borda de subida, denominada em inglês “positive one shot”, seta a “1” lógico um bit para um só ciclo de scan do PLC. Na operação contrária, a função de detecção da borda de descida, denominada em inglês “negative one shot”, seta a “1” lógico um bit para um só ciclo de scan do PLC. Veja a Figura 8.1.

DETECÇÃO DE BORDA DE DESCIDA E SUBIDA

8.0 Generalidades

8.1 Definição de Borda de Descida e Subida

8.2 Construção da Operação de Detecção da Borda de Descida e Subida

8.3 Operação Especial de Detecção com a CPU S7-1200

8.4 Aplicação: Partida e Parada com Segurança de um MAT (Motor Assíncrono Trifásico)

8.0 Generalidades

Em automação industrial, frequentemente é necessária a gestão de um sinal no momento em que este muda de estado, e não quando está em um estado definido. Essa gestão de mudança de estado de um sinal pode ser útil na lógica de controle de qualquer máquina industrial.

Ocorre, muitas vezes, que a lógica de controle precise apenas detectar a mudança de estado de um botão de comando, por exemplo, quando, por falha do operador, se pressiona de forma constante o botão de acionamento da máquina.

Esse procedimento pode causar uma movimentação perigosa da máquina. Se ao contrário detectamos a mudança de estado do botão, essa movimentação perigosa da máquina não acontece, tornando a instalação em condição de plena segurança.

8.1 Definição de Borda de Descida e Subida

Referimo-nos à borda de subida de um sinal elétrico quando este passa de um estado a “0” lógico a um estado a “1” lógico ou de off a on. Referimo-nos à borda de descida de um sinal elétrico quando este passa de um estado a “1” lógico a um estado a “0” lógico ou de on a off.

A função de detecção da borda de subida, denominada em inglês “positive one shot”, seta a “1” lógico um bit para um só ciclo de scan do PLC. Na operação contrária, a função de detecção da borda de descida, denominada em inglês “negative one shot”, seta a “1” lógico um bit para um só ciclo de scan do PLC. Veja a Figura 8.1.

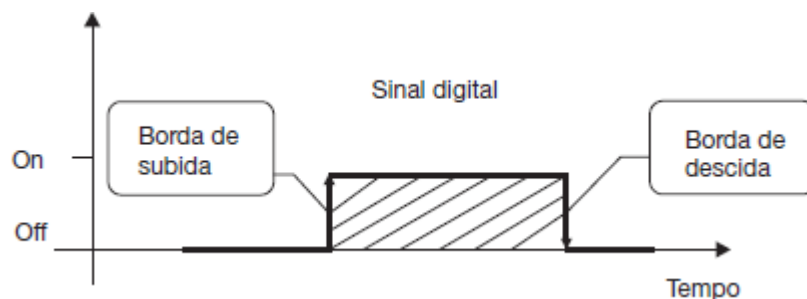


FIGURA 8.1

8.2 Construção da Operação de Detecção da Borda de Descida e Subida

Nem todos os PLCs possuem instruções especiais para a gestão da detecção da borda de descida e subida de um sinal. Nesse caso, com poucas linhas de programas é possível detectar a borda de descida e subida de um sinal. Na Figura 8.2A temos um exemplo de detecção de borda de subida.

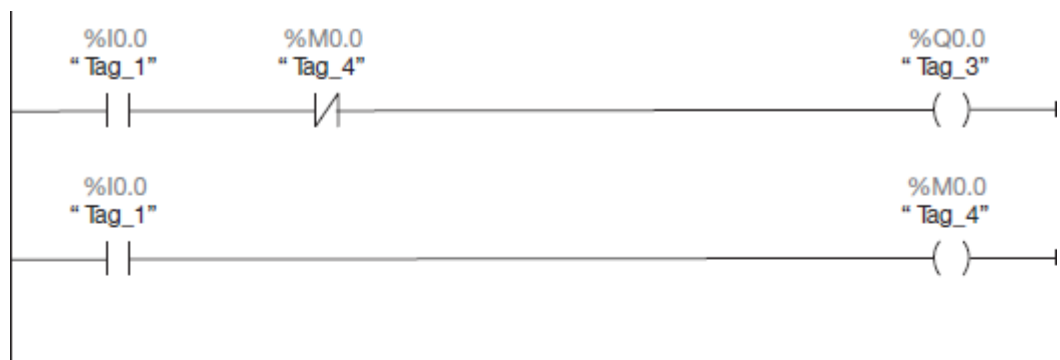
Pressionando o botão I0.0 (borda de subida) se energiza a saída Q0.0. Dado que o contato auxiliar de Merker M0.0 é normalmente fechado, simultaneamente se energiza a bobina do Merker M0.0. Quando termina o scan do PLC (poucos milissegundos), se retorna à primeira linha de programa. Nesse retorno do ciclo de scan, o contato auxiliar de Merker M0.0, que era normalmente fechado, se abre, desenergizando a saída Q0.0.

Em consequência, a saída Q0.0 é energizada somente para um tempo igual a um ciclo de scan do PLC.

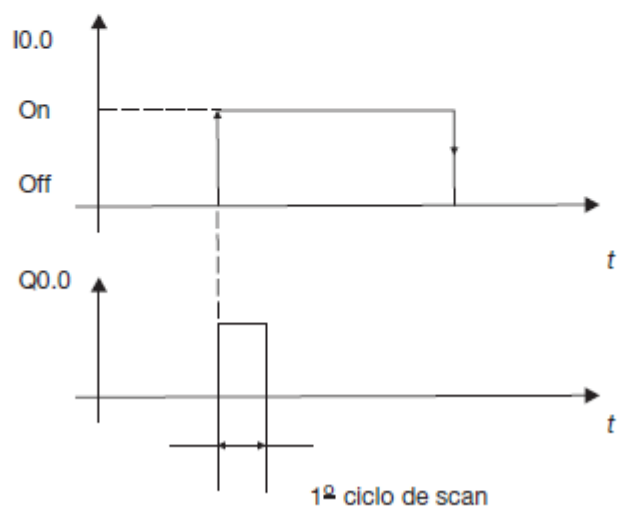
Na Figura 8.3A temos um pequeno programa em linguagem Ladder que detecta a borda de descida de um sinal.

Pressionando o botão I0.0 (borda de subida) se seta a bobina de Merker M0.1, fechando, assim, o contato auxiliar de Merker M0.1, que era normalmente aberto (primeira linha de programa).

A saída Q0.0 permanece desenergizada enquanto o botão I0.0 nesse momento é aberto, devido ao botão I0.0 pressionado. Relaxando o botão I0.0 (borda de descida) o botão retorna à situação inicial, energizando assim a saída Q0.0. Lembramos que dessa vez o contato auxiliar de Merker M0.1 está fechado.

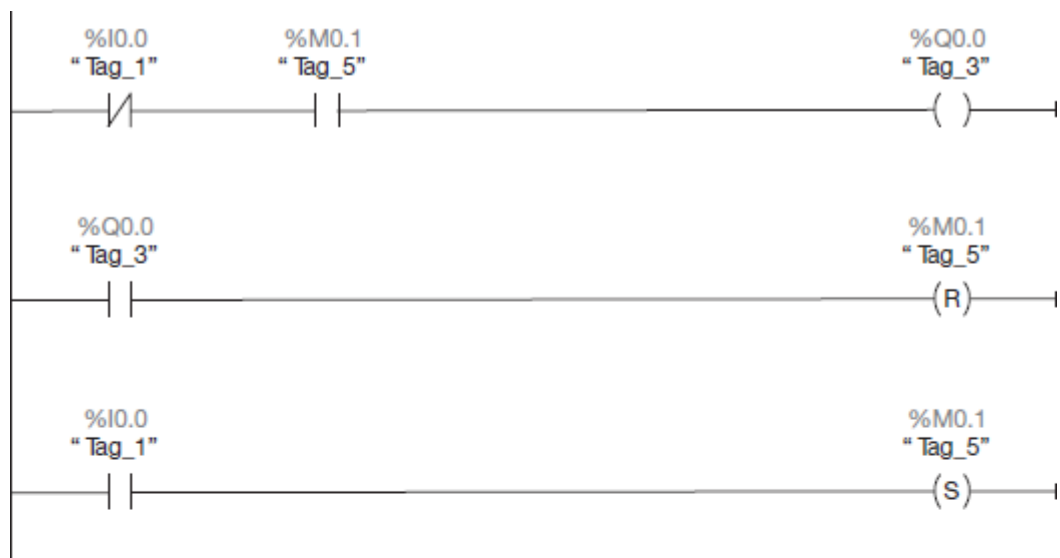


(A)

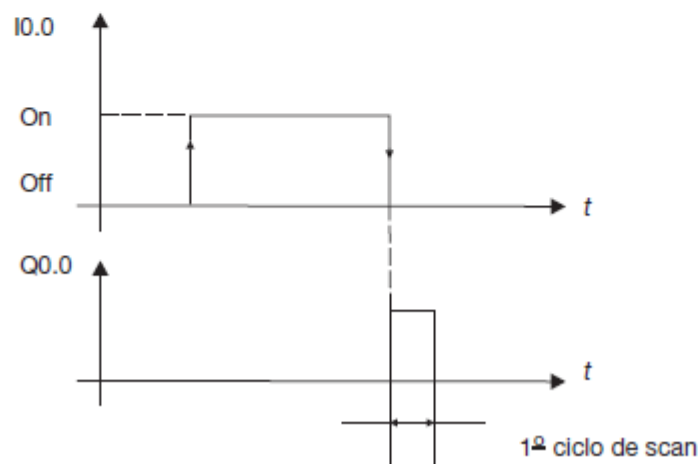


(B)

FIGURA 8.2



(A)



(B)

FIGURA 8.3

No seguinte ciclo de scan do PLC na 2ª linha de programa, o contato auxiliar de Q0.0 é fechado, “ressetando”, assim, o Merker M0.1 e, em consequência, também a saída Q0.0.

8.3 Operação Especial de Detecção com a CPU S7-1200

A CPU S7-1200 dispõe de duas instruções especiais para a detecção da borda de subida e descida de um sinal elétrico.

Essa instrução especial deve ser inserida na frente de uma saída qualquer tipo bobina ou boxe. A representação é do tipo a contato. Na Tabela 8.1 são representadas essas duas instruções especiais para a detecção da borda de subida e descida de um sinal elétrico com a CPU S7-1200.

TABELA 8.1 Tabela dos Símbolos

LAD	Função
	A operação “detecção de borda positiva” detecta em um ciclo de scan a transição de 0 a 1. O contato “transição positiva” permite que a corrente circule por um ciclo de scan a cada transição de 0 a 1. A variável IN é o bit em que deve ser detectada a borda de subida; a variável M_BIT é o Merker de armazenamento do estado anterior da entrada.
	A operação “detecção de borda negativa” detecta em um ciclo de scan a transição de 1 a 0. O contato “transição negativa” permite que a corrente circule por um ciclo de scan a cada transição de 1 a 0. A variável IN é o bit em que deve ser detectada a borda de descida; a variável M_BIT é o Merker de armazenamento do estado anterior da entrada.

Na Figura 8.4 temos a representação da Tabela 8.1 no equivalente da linguagem FBD.

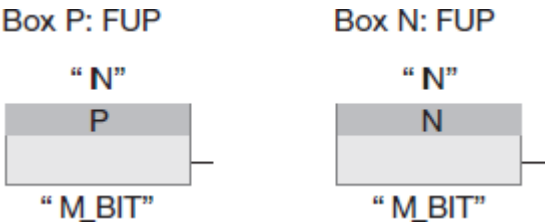


FIGURA 8.4

Na Figura 8.5 está representado o emprego dessa instrução com as linguagens Ladder e FBD.

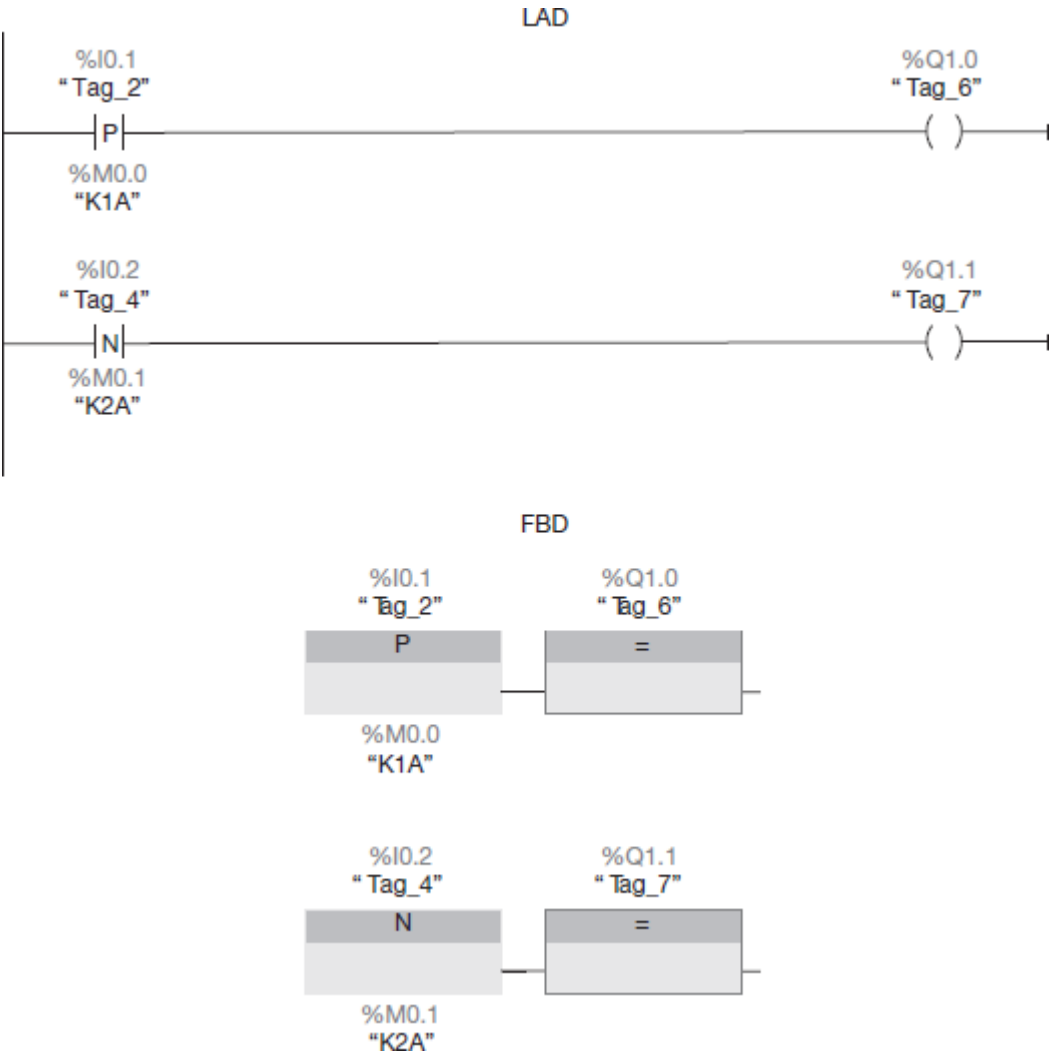


FIGURA 8.5

Na Figura 8.6 está representado o diagrama temporal relativo ao pequeno programa da Figura 8.5.

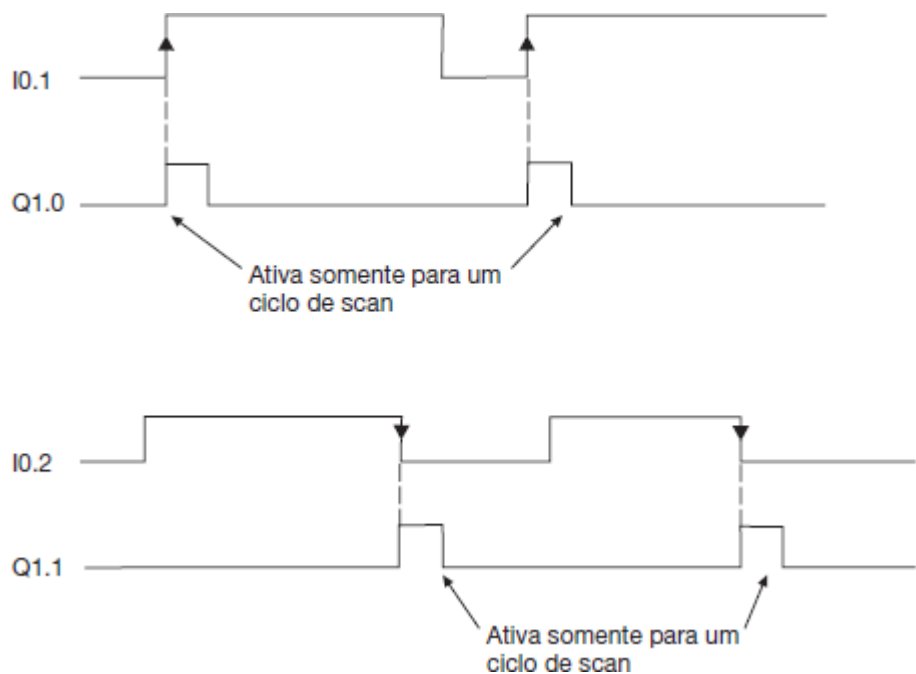


FIGURA 8.6

8.4 Aplicação: Partida e Parada com Segurança de um MAT (Motor Assíncrono Trifásico)

A aplicação a seguir demonstra como se programa uma simples partida e parada com segurança de um MAT, impedindo o operador de continuar pressionando o botão de partida. O pequeno programa é apresentado na Figura 8.7.

TABELA 8.2 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão de stop
S2	I0.1	Botão de start
FR	I0.2	Térmica motor
K1	Q0.0	Contator motor

Esquema Ladder e FBD da Partida e Parada em Segurança de um MAT (Motor Assíncrono Trifásico)

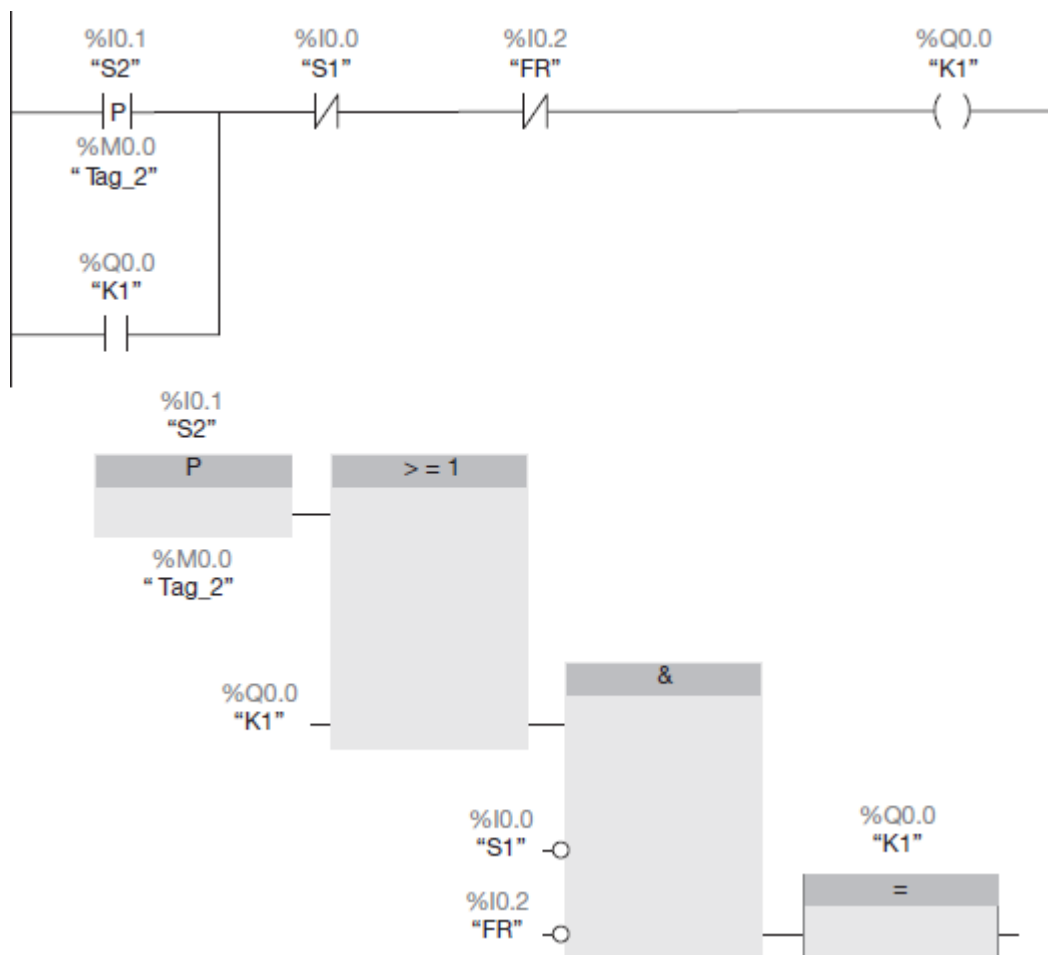


FIGURA 8.7

Nesse pequeno programa, suponhamos que o operador pressione constantemente o botão de start S2 e queira parar o motor pressionando o botão de stop S1.

No momento em que relaxa o botão de stop S1, ainda que mantenha constantemente pressionado o botão de start S2, o motor não torna a partir, fazendo com que a instalação fique em condições de plena segurança, graças à presença, no programa do contato, da transição positiva P. Sem a presença do contato P, ocorreria que, ao relaxar o botão de stop S1 e manter constantemente pressionado o botão de start S2, o motor partiria novamente, colocando a instalação em condição de perigo.

TEMPORIZADORES

TEMPORIZADORES

- 9.0 Generalidades
 - 9.1 Imposição por Meio do Programa de um Determinado Tempo de Atraso
 - 9.2 Timer de Pulso
 - 9.3 Timer com Atraso no Desligamento
 - 9.4 Timer com Atraso na Ligação com Memória
 - 9.5 Timer com Atraso na Ligação
 - 9.6 Operação de Temporização com a CPU S7-1200
 - 9.7 Energização de uma Saída com Timer TP
 - 9.8 Aplicação: Lubrificação de um Motor Elétrico de Grande Potência
 - 9.9 Aplicação: Circuito Eletropneumático Relativo a Pausa no Fim do Curso Fca1 de um Cilindro, Antes do Retorno Automático (Eletroválvula 5/2 Biestável e Cilindro de Duplo Efeito –DE)
 - 9.10 Aplicação: Utilização do Timer TONR
 - 9.11 Geração de Sinal de Duração Preestabelecida
 - 9.12 Aplicação: Esteira Transportadora com Lampejo
-

9.0 Generalidades

Em automação, muito frequentemente surge a necessidade de subordinar algumas ações no transcorrer do tempo ou então gerar sinais de comando de duração preestabelecida.

Para satisfazer a essas duas exigências, o PLC S7-1200 tem dois tipos de instrução de temporização:

- Imposição por meio do programa de um determinado tempo de atraso;
- Geração de sinal com duração preestabelecida.

9.1 Imposição por Meio do Programa de um Determinado Tempo de Atraso

TEMPORIZADORES

- 9.0 Generalidades
 - 9.1 Imposição por Meio do Programa de um Determinado Tempo de Atraso
 - 9.2 Timer de Pulso
 - 9.3 Timer com Atraso no Desligamento
 - 9.4 Timer com Atraso na Ligação com Memória
 - 9.5 Timer com Atraso na Ligação
 - 9.6 Operação de Temporização com a CPU S7-1200
 - 9.7 Energização de uma Saída com Timer TP
 - 9.8 Aplicação: Lubrificação de um Motor Elétrico de Grande Potência
 - 9.9 Aplicação: Circuito Eletropneumático Relativo a Pausa no Fim do Curso Fca1 de um Cilindro, Antes do Retorno Automático (Eletroválvula 5/2 Biestável e Cilindro de Duplo Efeito –DE)
 - 9.10 Aplicação: Utilização do Timer TONR
 - 9.11 Geração de Sinal de Duração Preestabelecida
 - 9.12 Aplicação: Esteira Transportadora com Lampejo
-

9.0 Generalidades

Em automação, muito frequentemente surge a necessidade de subordinar algumas ações no transcorrer do tempo ou então gerar sinais de comando de duração preestabelecida.

Para satisfazer a essas duas exigências, o PLC S7-1200 tem dois tipos de instrução de temporização:

- Imposição por meio do programa de um determinado tempo de atraso;
- Geração de sinal com duração preestabelecida.

9.1 Imposição por Meio do Programa de um Determinado Tempo de Atraso

Como já citamos anteriormente, os dispositivos mais usados no desenvolvimento da lógica de controle do processo, além das bobinas e dos contatos, são os temporizadores (em inglês, *timer*).

O timer mais comum é o temporizador com atraso na ligação. Dele derivam todas as outras funções de temporizadores, entre esses o temporizador com atraso no desligamento, o temporizador de pulso, o temporizador múltiplo e outros.

Geralmente em um micro PLC existem dois ou três tipos de temporizadores. Esses tipos de timer podem ser combinados de várias modalidades, utilizando os respectivos contatos para realizar tarefas de automação variadas.

O exemplo que daremos a seguir é válido para todos os tipos de PLC. Os quatro tipos de timer mais utilizados são:

- timer de pulso;
- timer com atraso no desligamento;
- timer com atraso na ligação com memória;
- timer com atraso na ligação.

9.2 Timer de Pulso

Esse tipo de temporizador, em inglês *pulse timer*, é o equivalente de um circuito monoestável, ou seja, gera um pulso cuja duração é definida por meio da entrada I0.1 (entrada de preset).

Com a borda de subida da entrada I0.1, o pulso de duração preestabelecida de 8 segundos parte, independentemente do estado da entrada I0.1.

Na Figura 9.1, temos o diagrama elétrico desse timer, enquanto na Figura 9.2 temos o diagrama Ladder equivalente e o andamento da entrada I0.1 e Q0.1 em função do tempo.

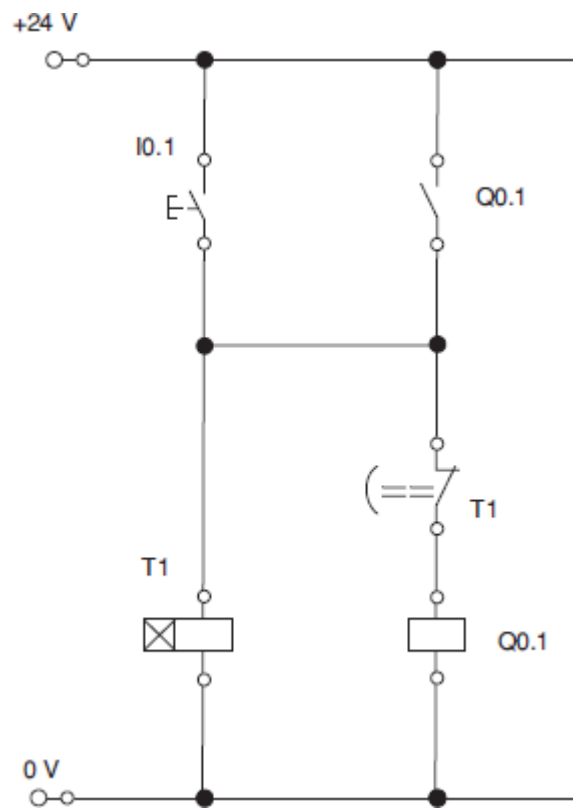


FIGURA 9.1

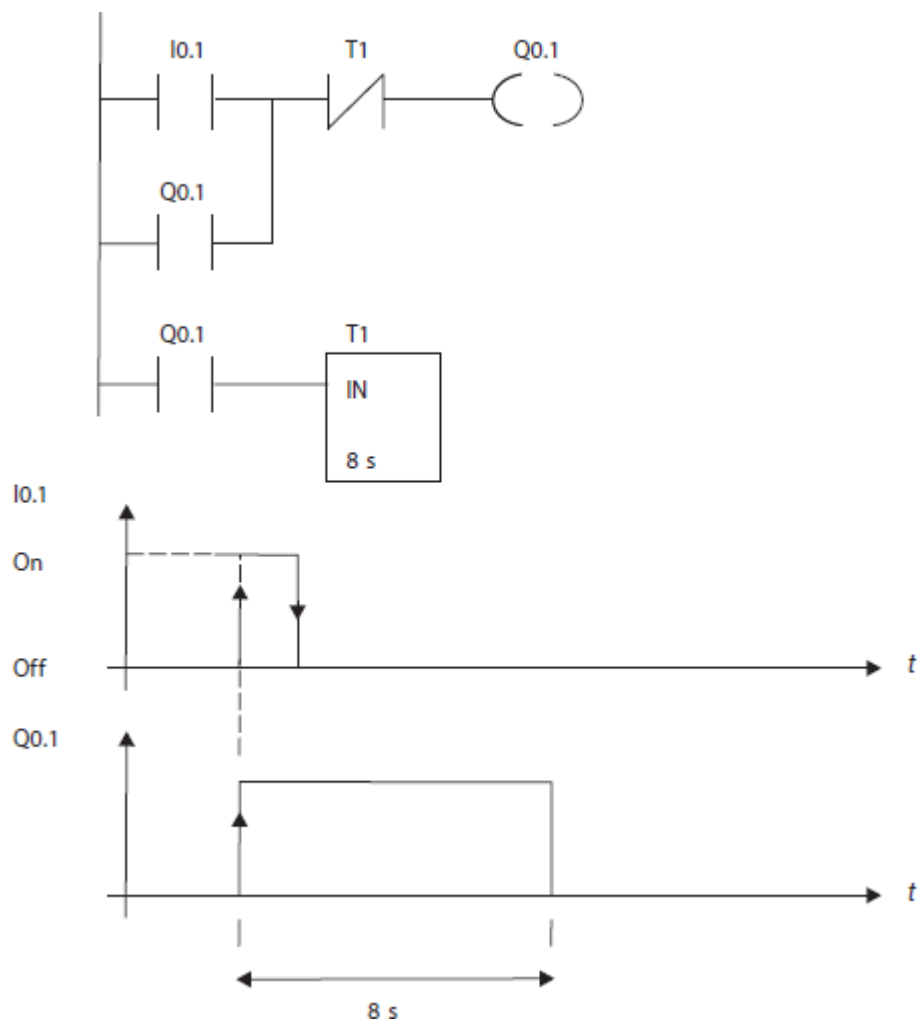


FIGURA 9.2

O diagrama elétrico do timer com atraso no desligamento acionado por uma única entrada é indicado na Figura 9.3; na Figura 9.4 temos o diagrama Ladder equivalente.

O funcionamento é muito simples: quando a entrada funcionando como chave I0.0 se ativa (se fecha), o contato do temporizador T1 se fecha, ativando assim a saída Q0.0. Quando a mesma entrada I0.0 se abre (off), inicia-se a temporização. Transcorrido o tempo de 9 segundos, o contato T1 se abre e a saída Q0.0 se desativa. Se no intervalo do tempo entre 0 e 9 segundos a entrada I0.0 se fecha (on), o timer zera, mas o contato T1 permanece fechado, portanto a saída Q0.0 fica ativa (on). Na Figura 9.5 é demonstrado o andamento da entrada I0.0 e Q0.0 em função do tempo.

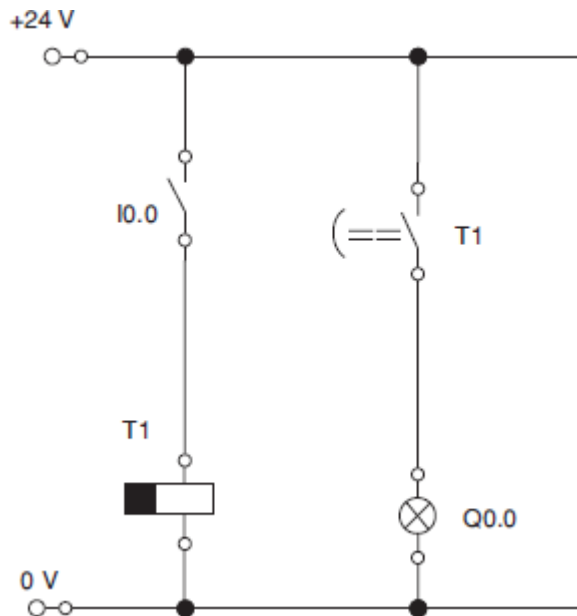


FIGURA 9.3

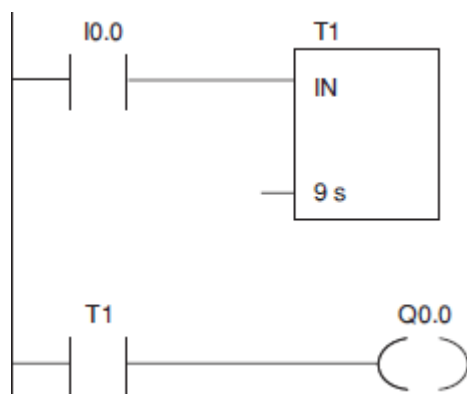


FIGURA 9.4

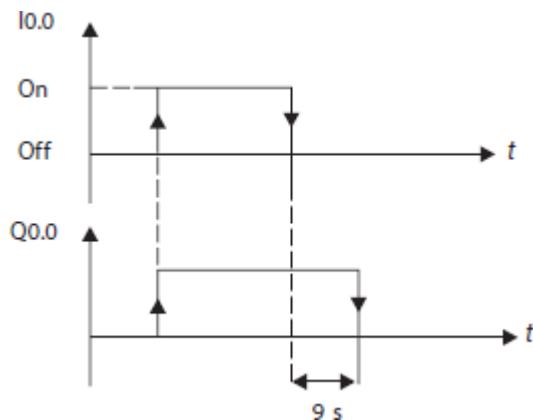


FIGURA 9.5

9.4 Timer com Atraso na Ligação com Memória

O timer com atraso na ligação com memória acionado por uma única entrada é indicado na Figura 9.6.

O funcionamento desse timer é o seguinte: quando a entrada I0.1 se ativa (on), inicia-se a contagem do tempo, exatamente como no temporizador com atraso na ligação. Transcorrido o tempo, ou seja, os 10 segundos estabelecidos, se fecha o contato T1 e a saída Q0.0

se ativa (on).

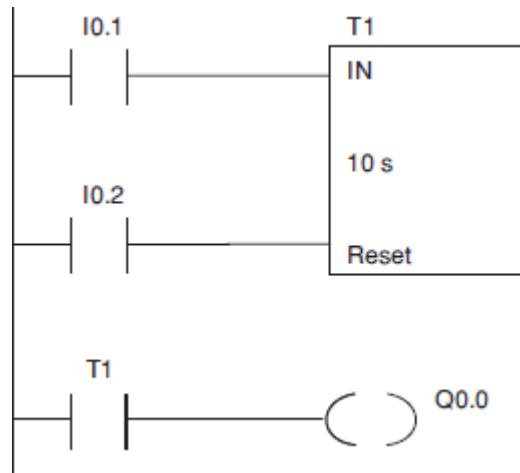


FIGURA 9.6

A diferença principal que existe em relação ao temporizador com atraso na ligação é que ele possui uma memória do tempo transcorrido. Vejamos um exemplo: se no intervalo de tempo entre 0 e 10 segundos a entrada I0.1 for aberta (off), o timer memoriza o tempo transcorrido até aquele momento. Supondo que tenha memorizado 5 segundos, no ato do fechamento seguinte da entrada I0.1 (on) o tempo contado parte de 5 segundos até chegar a 10 segundos, ativando assim a saída Q0.0. Quando a entrada I0.2 se ativa (on), se resseta a contagem do tempo em qualquer momento. Na Figura 9.7 é demonstrado o andamento da entrada I0.1, I0.2 e Q0.0 em função do tempo.

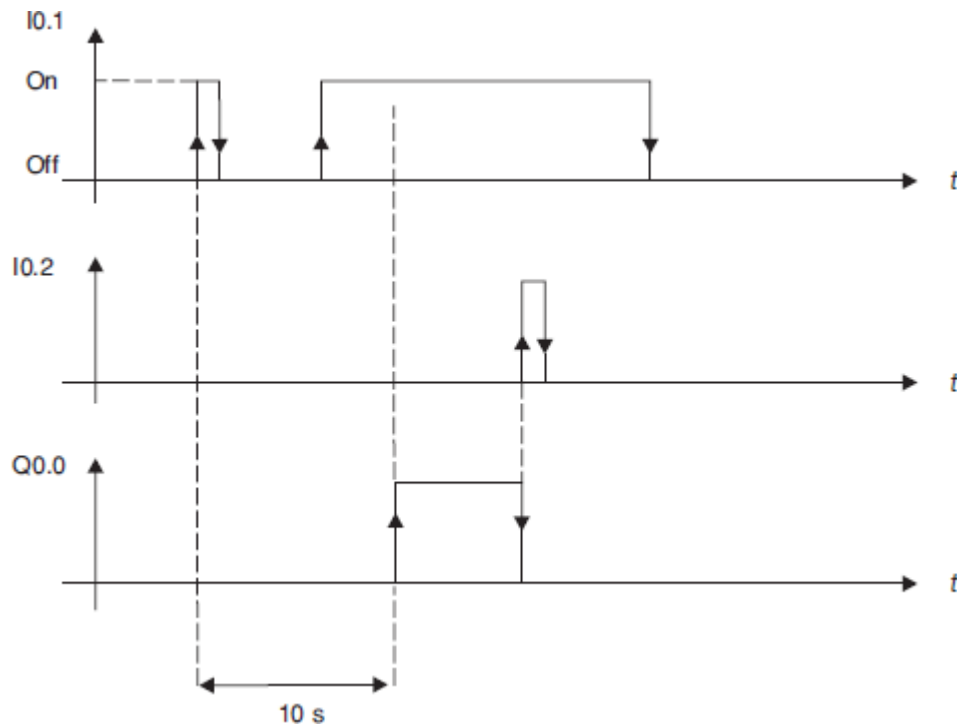


FIGURA 9.7

9.5 Timer com Atraso na Ligação

O timer com atraso na ligação acionado por uma única entrada é indicado na Figura 9.8. Na Figura 9.9 é representado o diagrama Ladder equivalente.

O funcionamento é o seguinte: quando a entrada I0.0 se ativa (on), iniciase a contagem do tempo. Transcorrido esse tempo, ou seja, os 5 segundos estabelecidos, se fecha o contato T1 e a saída Q0.0 se ativa (on). Em seguida, quando a entrada I0.0 se abre o timer zera, o contato T1 se abre e a saída Q0.0 se desativa (off).

Se no intervalo de tempo entre 0 e 5 segundos a entrada I0.0 se abre, o timer zera. Na Figura 9.10 é demonstrado o andamento da entrada I0.0, Q0.0 em função do tempo.

9.6 Operação de Temporização com a CPU S7-1200

A CPU S7-1200 dispõe de quatro operações de temporização diferentes, já amplamente discutidas nas seções anteriores:

- Temporização de pulso (**TP**)
- Temporização com atraso na ligação (**TON**)
- Temporização com atraso no desligamento (**TOF**)
- Temporização com atraso na ligação com memória (**TONR**)

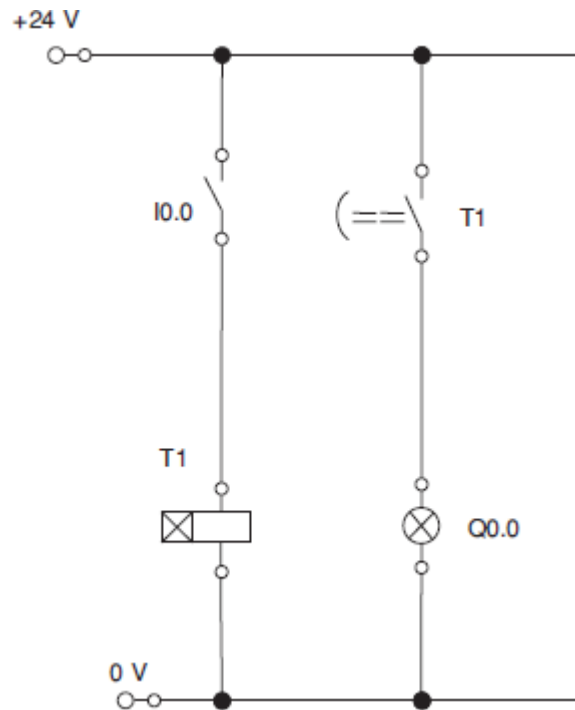


FIGURA 9.8

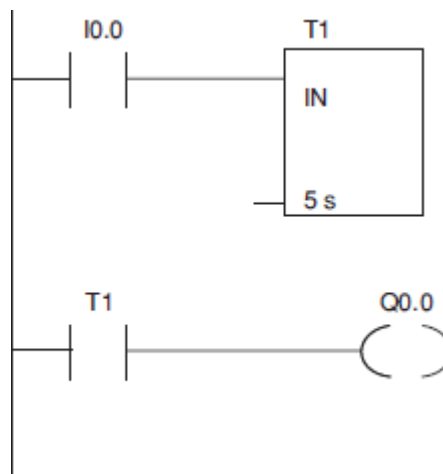


FIGURA 9.9

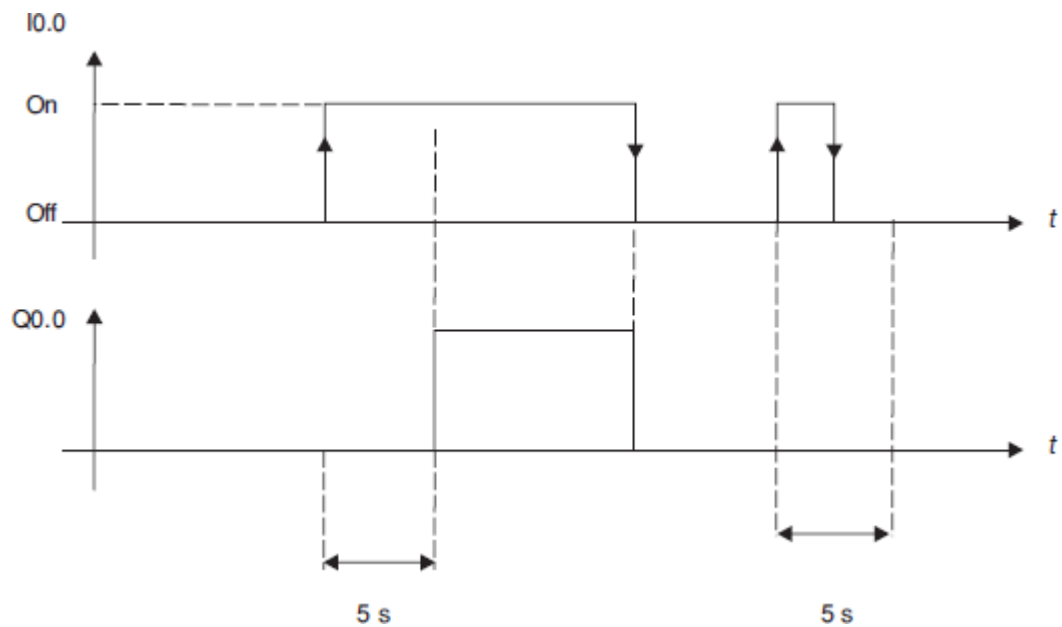


FIGURA 9.10

9.6.1 Timers TP, TON, TOF, TONR

Nas CPUs S7-1200 os boxes dos timers TP, TON, TOF apresentam-se conforme Figura 9.11.

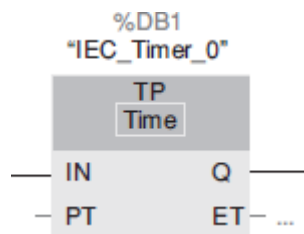


FIGURA 9.11

O timer TONR (veja a Figura 9.12) apresenta ainda a entrada de reset contagem (R).

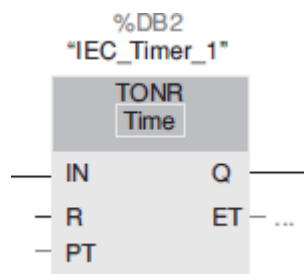


FIGURA 9.12

A bobina ----[RT]---- resseta todos os dados dos temporizadores a qualquer momento.

A Tabela 9.1 apresenta o significado dos parâmetros dos timers.

TABELA 9.1 Tabela dos Símbolos

Parâmetro	Tipo de dados	Comentário
IN	Bool	Ativa a entrada do timer
R	Bool	Zera a contagem
PT	Time	Imposta valor do tempo
Q	Bool	Saída do timer
ET	Time	Saída do tempo transcorrido
Blocos dados do timer	DB	Especifica o timer

9.6.2 Base dos Tempos

Os valores do tempo imposto PT e do tempo transcorrido ET são armazenados como números inteiros de 32 bits com sinal, representados em milissegundos. Os dados TIME utilizam o formato T# para serem especificados e podem sê-lo como unidade de tempo simples “T#150ms” ou composta “T#2s_200ms”.

O tempo máximo que pode ser armazenado no timer é T#24d_20h_31m_23s_647ms .

9.6.3 Exemplos de TP, TON, TOF, TONR

– Timer TP

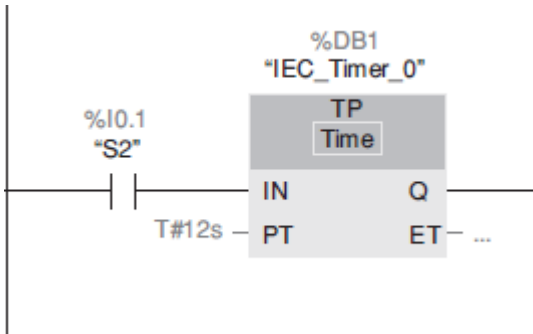


FIGURA 9.13

O timer TP apresenta um tempo imposto simples de 12 segundos (12 s).

– Timer TON

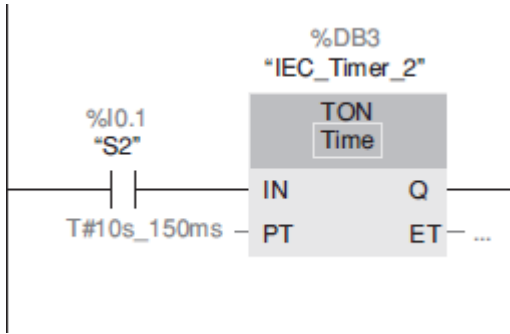


FIGURA 9.14

O timer TON apresenta um tempo composto de 10 segundos e 150 milissegundos.

– Timer TOF

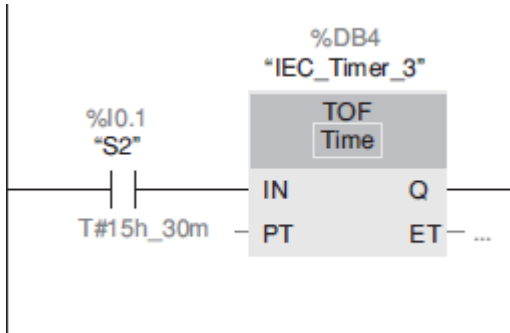


FIGURA 9.15

O timer TOF apresenta um tempo composto de 15 horas e 30 minutos.

– Timer TONR

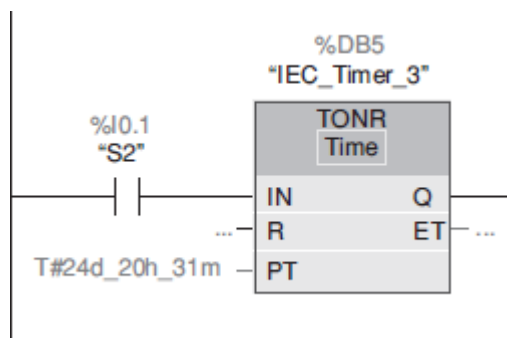


FIGURA 9.16

O timer TONR apresenta um tempo composto de 24 dias, 20 horas e 31 minutos.

9.7 Energização de uma Saída com Timer TP

Para energizar uma saída com os timers do S7-1200, é aconselhável armazenar anteriormente o bit de saída timer Q na memória temporária tipo Merker.

Vejamos um exemplo.

Após um pulso na entrada I0.4, queremos energizar a saída Q0.3 por um tempo de 20 segundos, independentemente do estado dessa entrada I0.4. O timer aconselhado é o TP, dado que gera um sinal de duração preestabelecida (veja a Figura 9.17).

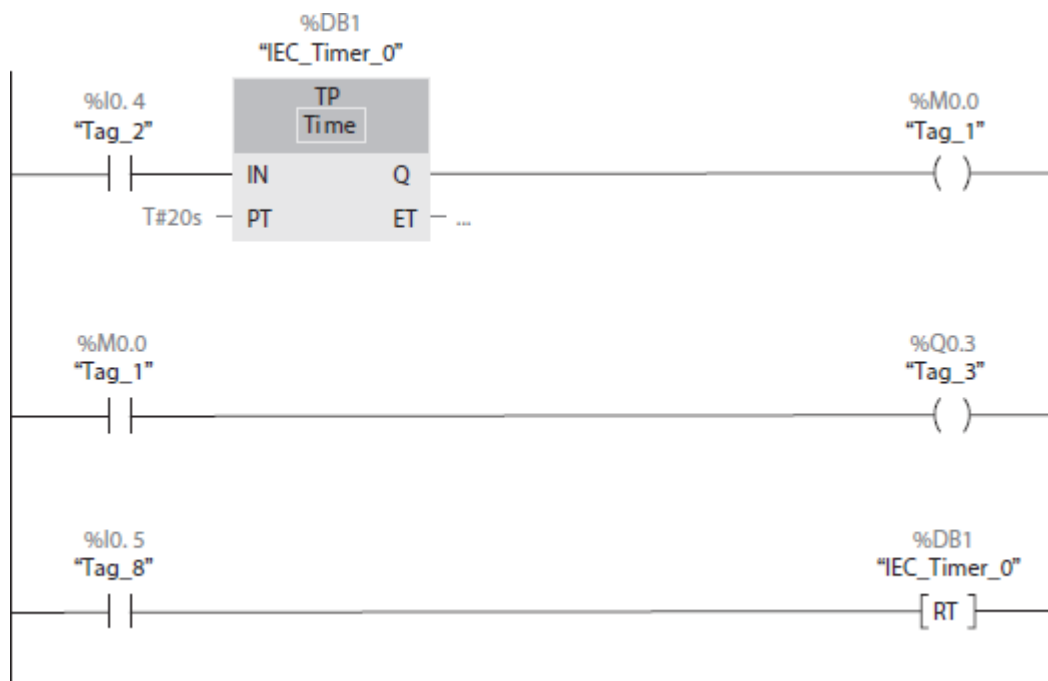


FIGURA 9.17

Notamos logo que a saída do timer Q é armazenada na bobina Merker M0.0. Após 20 segundos, temos o fechamento do contato Merker M0.0 com a energização da saída discreta Q0.3.

A entrada I0.5 resseta o timer a qualquer momento por meio da bobina RT.

9.8 Aplicação: Lubrificação de um Motor Elétrico de Grande Potência

Um motor e a bomba de lubrificação do motor devem funcionar simultaneamente. Durante a fase de parada do motor, é solicitada a lubrificação da parte mecânica. Então, uma vez desligado o motor, a bomba de lubrificação deve continuar a funcionar por todo o intervalo de tempo necessário até que o motor pare definitivamente. Neste exemplo, a bomba permanece em funcionamento por outros 25 segundos depois que o motor é desligado. O esquema funcional resolutivo do problema é apresentado na Figura 9.18.

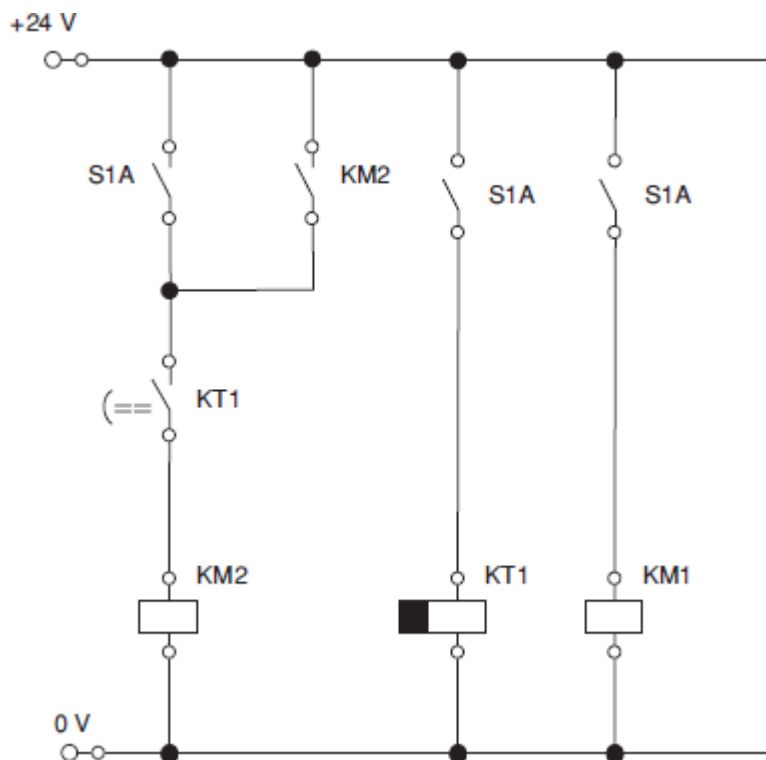


FIGURA 9.18

O esquema funcional da Figura 9.18 é muito simples. Quando o interruptor S1A está fechado, fecham-se simultaneamente as três ramificações do esquema funcional, temos assim a energização da bobina KM1 relativa ao motor de potência que começa a funcionar. Como consequência se energizará a bobina do relé retardado no desligamento KT1 que fecha imediatamente o seu contato auxiliar KT1, energizando assim a bobina KM2 relativa à bomba de lubrificação levando-a a começar a funcionar. Quando o interruptor S1A se abre KM1 se desenergiza parando assim o motor de potência. Neste momento se desenergiza também a bobina do temporizador KT1, assim inicia a contagem do tempo. Neste meio-tempo KM2 continua a funcionar enquanto o contato auxiliar de KT1 é fechado. Terminado o tempo pré-imposto, ou seja, os 25 segundos, o contato auxiliar de KT1 se abre parando assim a bomba de lubrificação KM2.

TABELA 9.2 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1A	I0.0	Interruptor de partida do motor
KM1	Q0.0	Contator do motor de potência
KM2	Q0.1	Contator da bomba de lubrificação
KT1	M0.0	Saída timer com atraso no desligamento

Diagrama Ladder resolutivo da Figura 9.18 (veja a Figura 9.19).

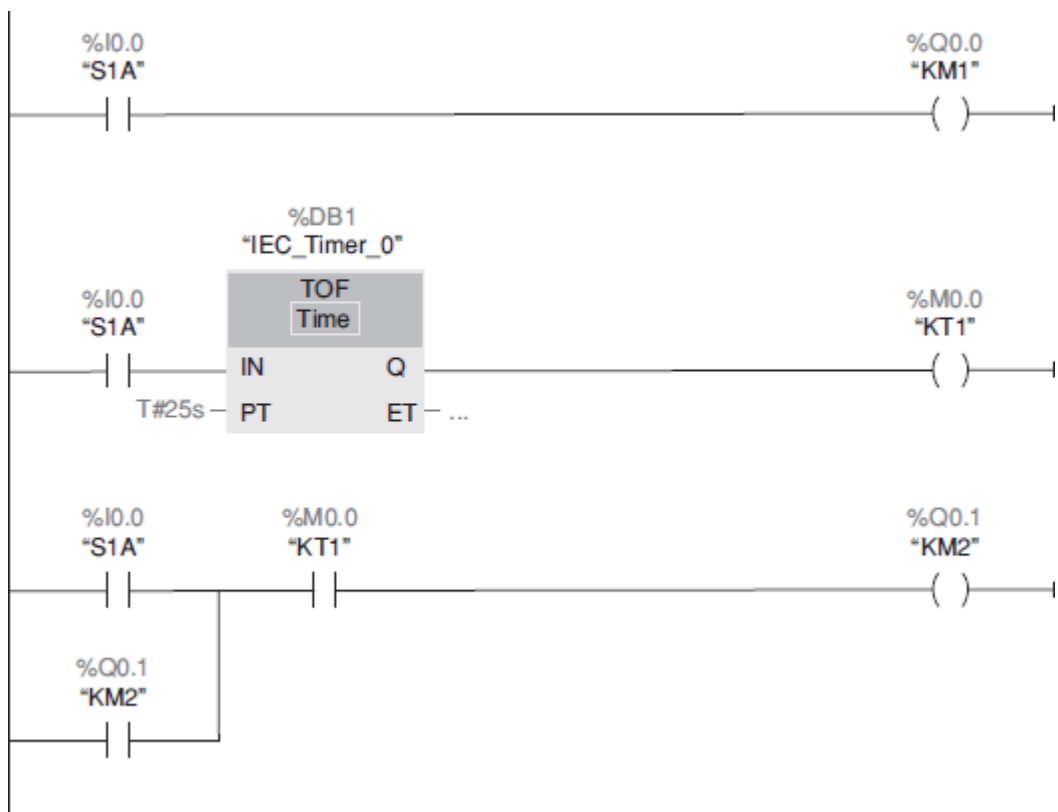


FIGURA 9.19

Diagrama FBD da Figura 9.19 (veja a Figura 9.20).

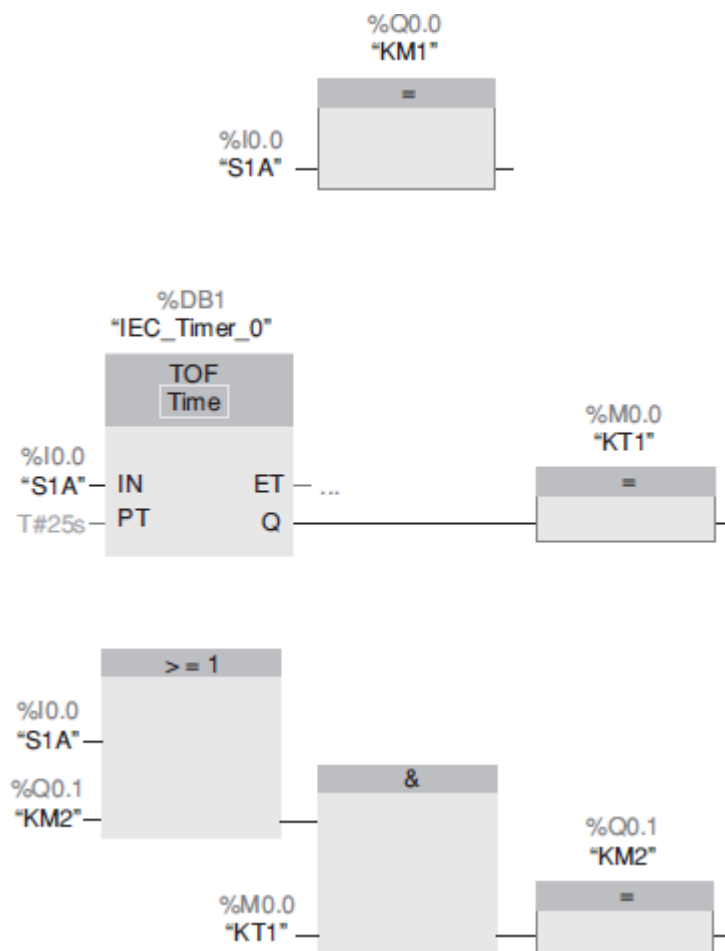


FIGURA 9.20

9.9 Aplicação: Circuito Eletropneumático Relativo a Pausa no Fim do Curso Fca1 de um Cilindro, Antes do Retorno Automático (Eletroválvula 5/2 Biestável e Cilindro de Duplo Efeito – DE)

O circuito da Figura 9.22 determina a pausa do cilindro no fim do curso Fca1 antes do retorno automático. Acionando o botão S1 se energiza a bobina do relé auxiliar KA1 e de Y1 (eletroválvula de saída do cilindro), e em consequência o cilindro sai. O contato do relé auxiliar KA1 de autorretenção se fecha, porém outros contatos KA1 em série a Y2 (eletroválvula de retorno do cilindro) se abrem. O cilindro que sai tocará o fim do curso Fca1, determinando assim a parada do cilindro. Começa também a contagem do temporizador KT1. Transcorrido o tempo de pausa, o contato do temporizador KT1 se abre, determinando a desenergização da bobina do relé auxiliar KA1 e de Y1. O contato auxiliar KA1 em série a Y2 se fecha, determinando a energização da eletroválvula Y2 e consequentemente o recuo do cilindro.

TABELA 9.3 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão partida
KT1	M0.0	Saída timer com atraso na ligação
Fca1	I0.1	Fim de curso cilindro fora
KA1	M0.1	Merker de relé auxiliar
Y1	Q0.0	Eletroválvula saída cilindro
Y2	Q0.1	Eletroválvula recuo cilindro

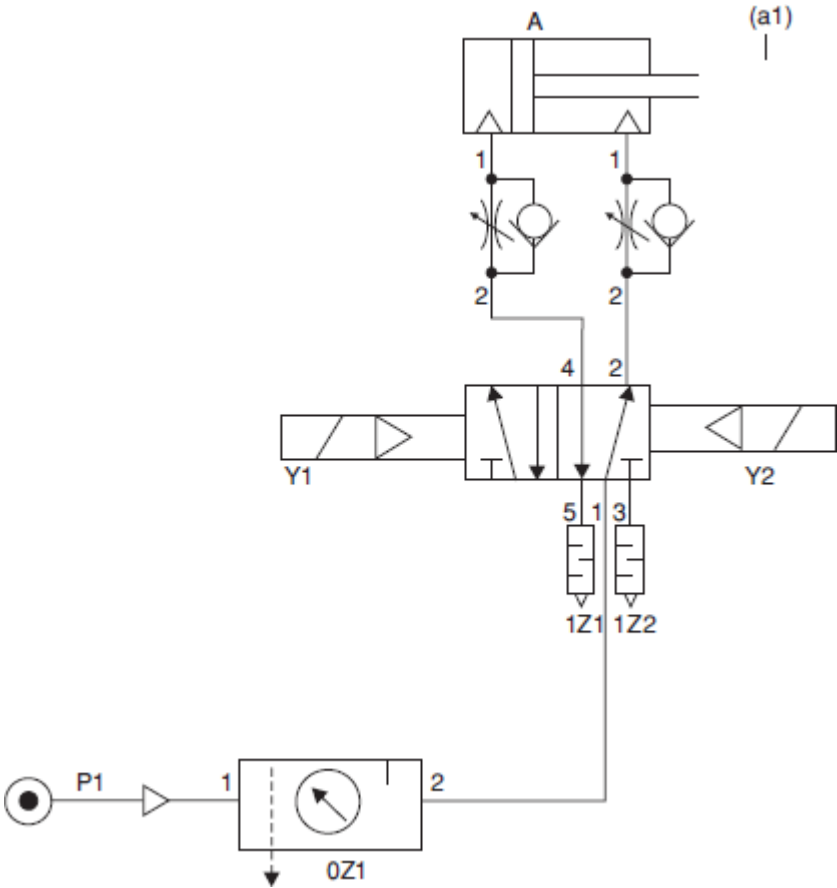


FIGURA 9.21 Esquema de potência do circuito eletropneumático.

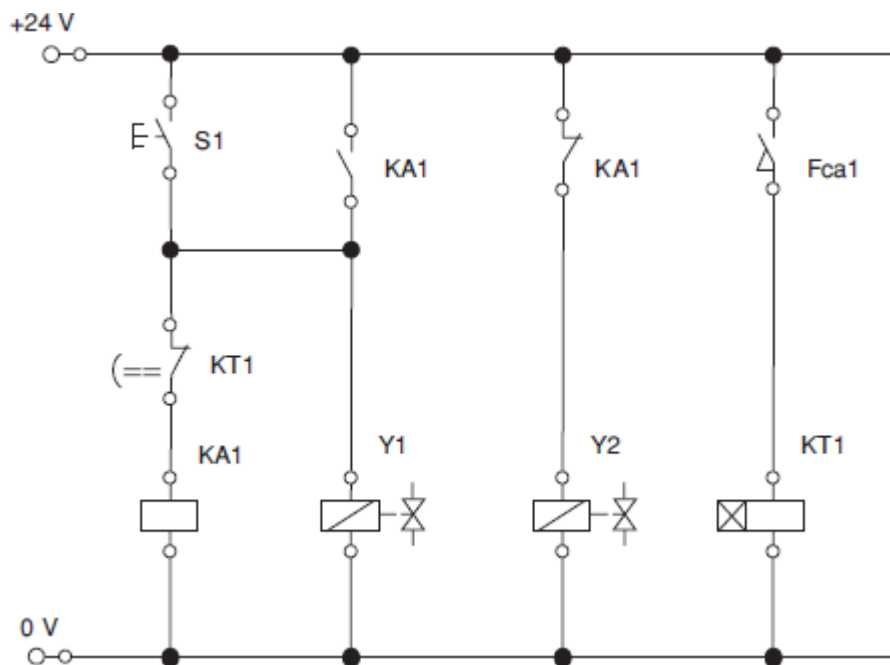


FIGURA 9.22

Diagrama Ladder resolutivo da Figura 9.22 (veja a Figura 9.23).

9.10 Aplicação: Utilização do Timer TONR

Pretende-se acionar um contator KM que ativa uma máquina operatriz depois de 30 segundos. Se por qualquer motivação a chave S1 se abre, o ato de refechamento dessa mesma chave S1 faz repartir o temporizador de onde a contagem do tempo foi interrompida até chegar aos 30 segundos estabelecidos. Com a chave S2 se resseta tudo em qualquer momento. O esquema Ladder resolutivo é mostrado na Figura 9.24. Esse esquema é muito simples e não precisa de mais explicações.

Na Figura 9.25 temos o diagrama FBD equivalente à Figura 9.24.

9.11 Geração de Sinal de Duração Prestabelecida

Veremos agora como é possível gerar sinais de duração preestabelecida com a CPU S7-1200.

Na CPU S7-1200 existe o conceito de **Merker de clock**, ou seja, um merker que consegue modificar periodicamente o seu estado binário com o valor pulso/pausa de 1:1.

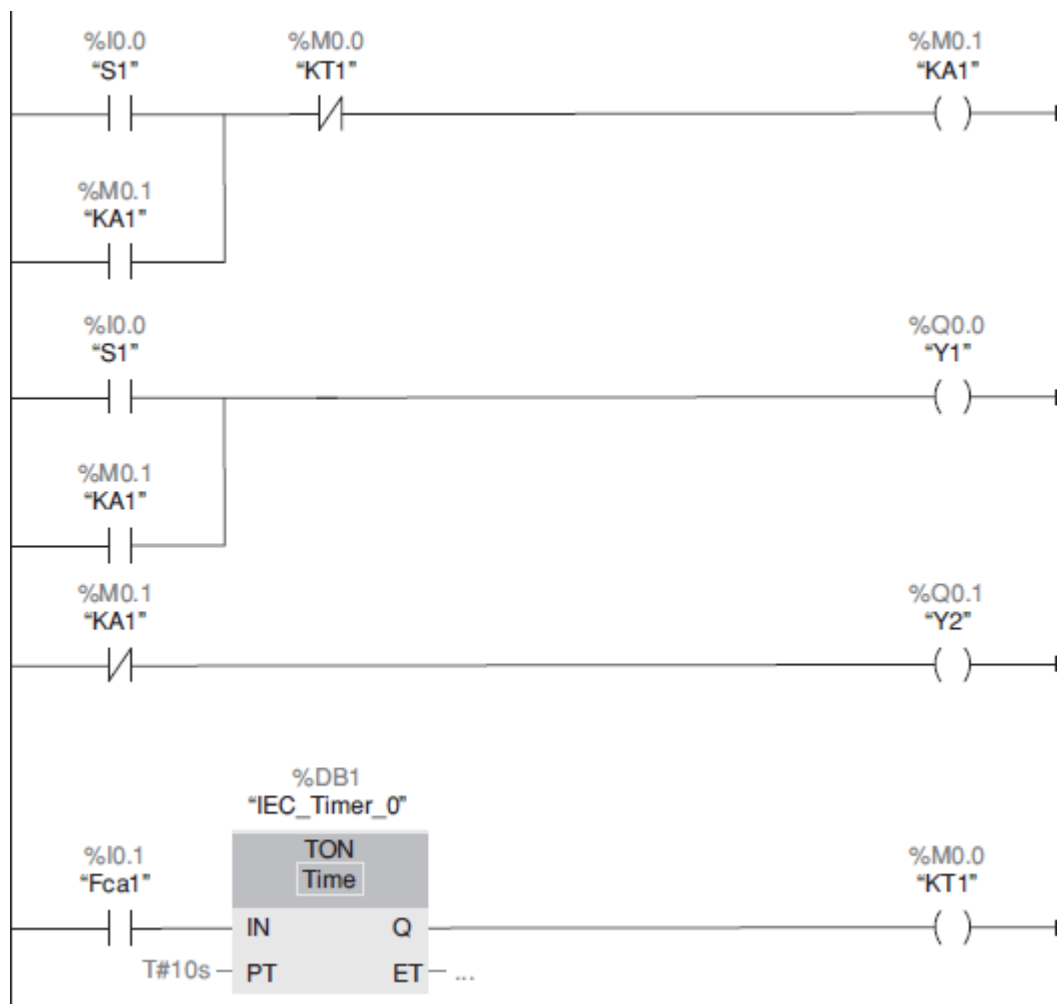


FIGURA 9.23

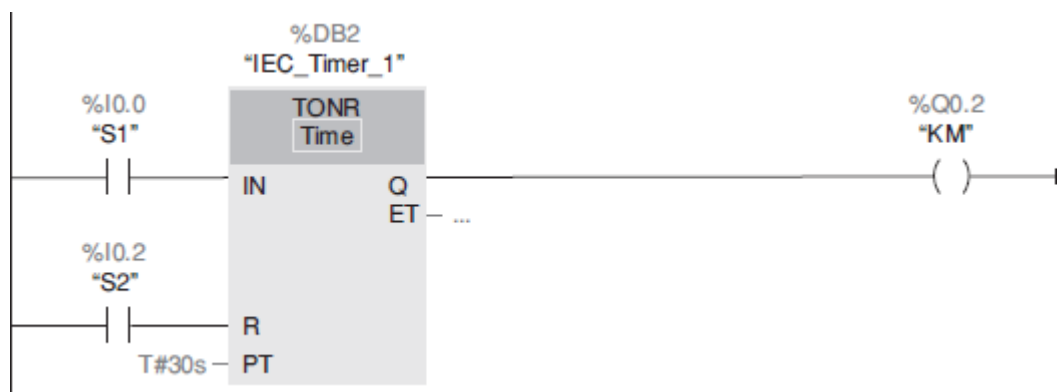


FIGURA 9.24

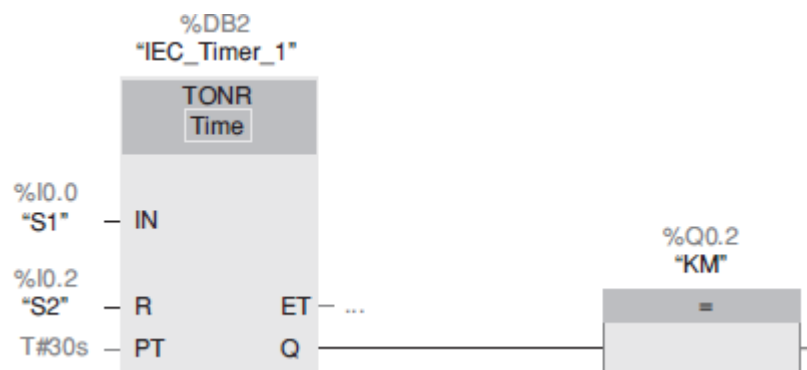


FIGURA 9.25

Para determinar qual byte (8 bits) de Merker da CPU deve ser um byte Merker de clock, utiliza-se a configuração Hardware do sistema operacional STEP 7 Basic, conforme a Figura 9.26.

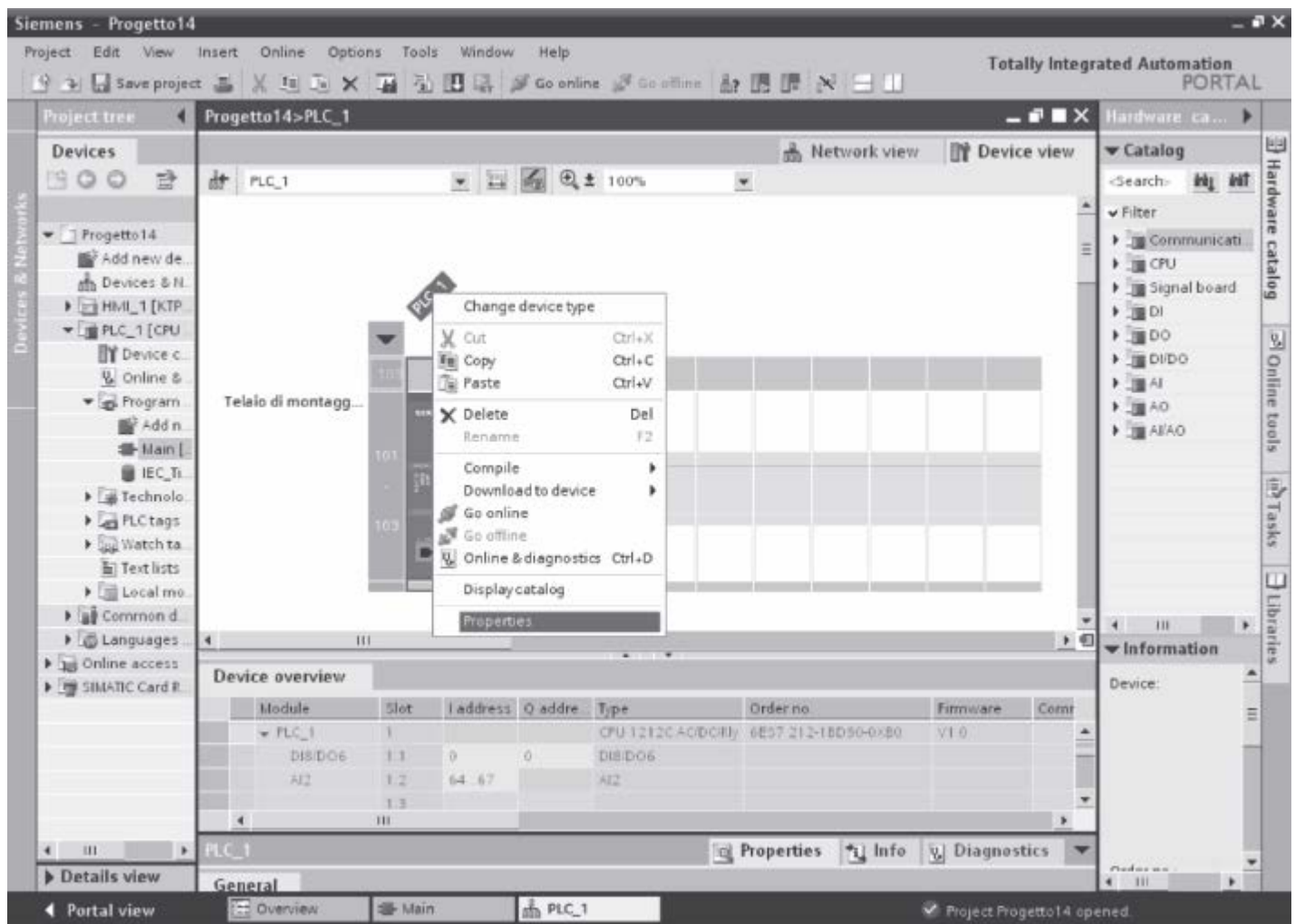


FIGURA 9.26

Basta clicar com o mouse à direita sobre o símbolo de cor azul de PLC-1 e selecionar **“Properties”**; logo aparece uma janela, como indicado na Figura 9.27.

Na Figura 9.27, a variação do byte Merker de clock (byte clock memory) com o valor de 0 (zero) significa que o byte Merker de clock é MB0. Selecionando o valor 1, o byte Merker de clock seria MB1, com o valor 2 seria MB2, e assim sucessivamente.

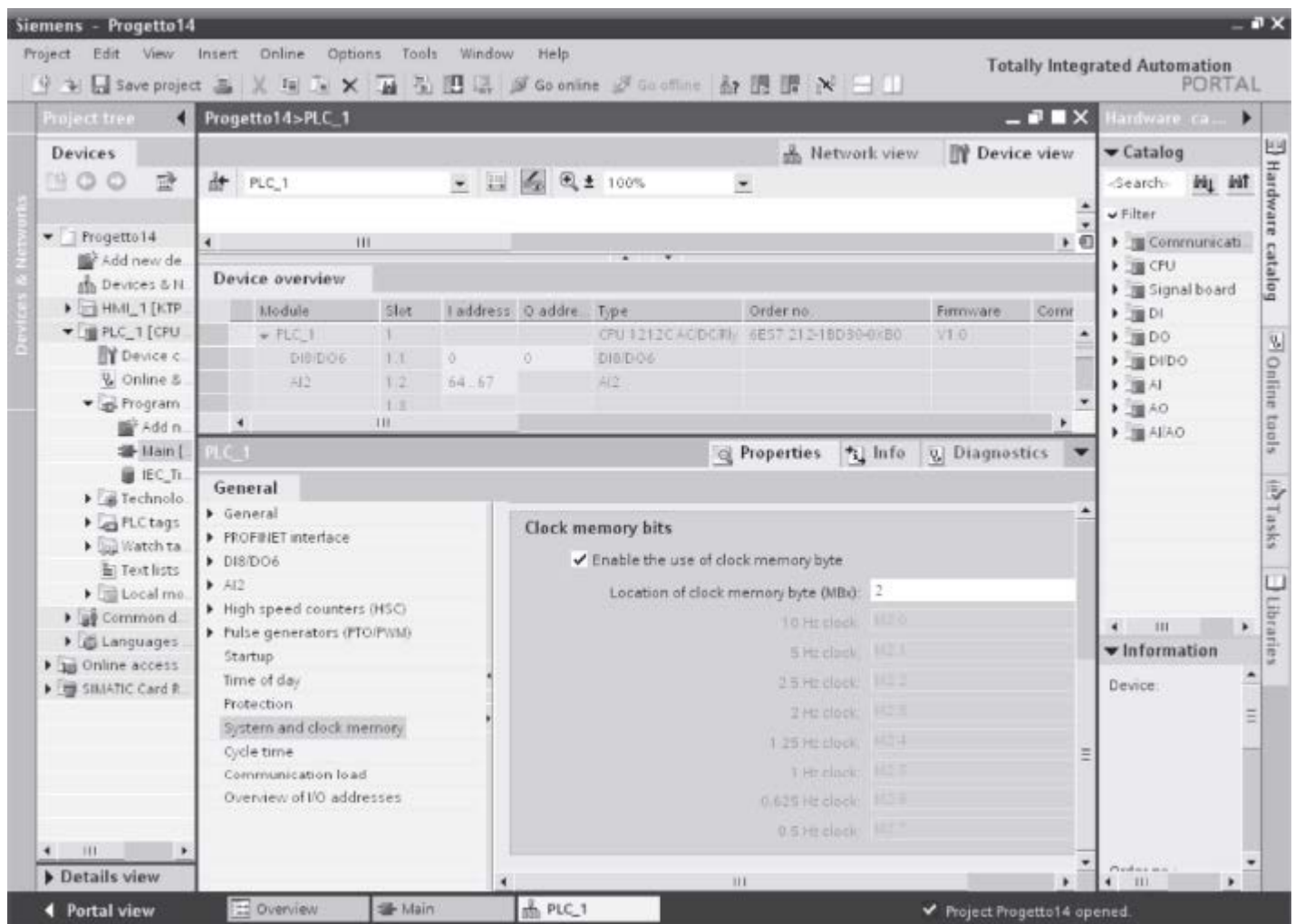


FIGURA 9.27

Conforme a Figura 9.27, foi selecionado o valor 2, então o byte de clock é MB2.

A cada bit do byte Merker de clock é definida uma frequência, conforme mostra a Figura 9.28. Esses valores não são modificáveis. Vejamos a seguir um exemplo prático de como se deve entender a Figura 9.28.

Suponhamos que a escolha do nosso byte Merker de clock seja MB2 definida como na Figura 9.27.

Bit do byte merker de clock	7	6	5	4	3	2	1	0
Duração (S)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frequência (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

FIGURA 9.28

Isso significa que os bits do byte MB2, ou seja, de M2.0 a M2.7, teriam, cada um, a duração e a frequência indicadas na Figura 9.29.

	M2.7	M2.6	M2.5	M2.4	M2.3	M2.2	M2.1	M2.0
Bit do byte merker de clock	7	6	5	4	3	2	1	0
Duração (S)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frequência (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

FIGURA 9.29

Por exemplo, o Merker M2.3 tem uma duração fixa de 0,5 s on/0,5 s off, enquanto o Merker M2.7 tem uma duração fixa de 2,0 s on/2,0 s off.

9.12 Aplicação: Esteira Transportadora com Lampejo

Uma esteira transportadora é acionada pressionando-se o botão S1 (veja a Figura 9.30). Inicia-se assim um lampejo de uma lâmpada

(H1) por 12 segundos. Depois se ativa um contator KM1 que aciona a esteira transportadora. Para a ativação do lampejo se escolhe o byte Merker de clock MB2, tendo M2.3 duração de 0,5 s off/0,5 s on.

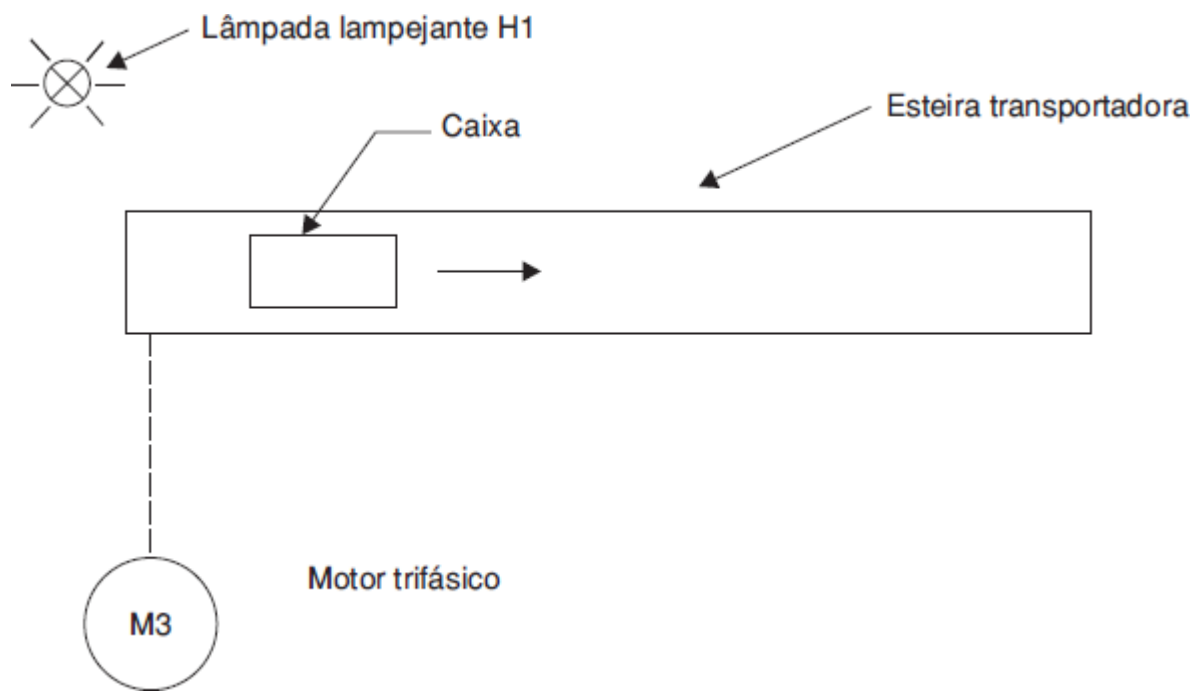


FIGURA 9.30

Tabela 9.4 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão de partida
KT1	M0.0	Saída timer TON
S2	I0.1	Botão de parada
K1A	M0.1	Merker de relé auxiliar
H1	Q0.0	Lâmpada
Clock_05	M2.3	Bit 0,5 soff/0,5 son
K1M	Q0.1	Contator de motor de esteira transportadora

Diagrama Ladder resolutivo da esteira transportadora com lampejo da Figura 9.30 (veja a Figura 9.31).

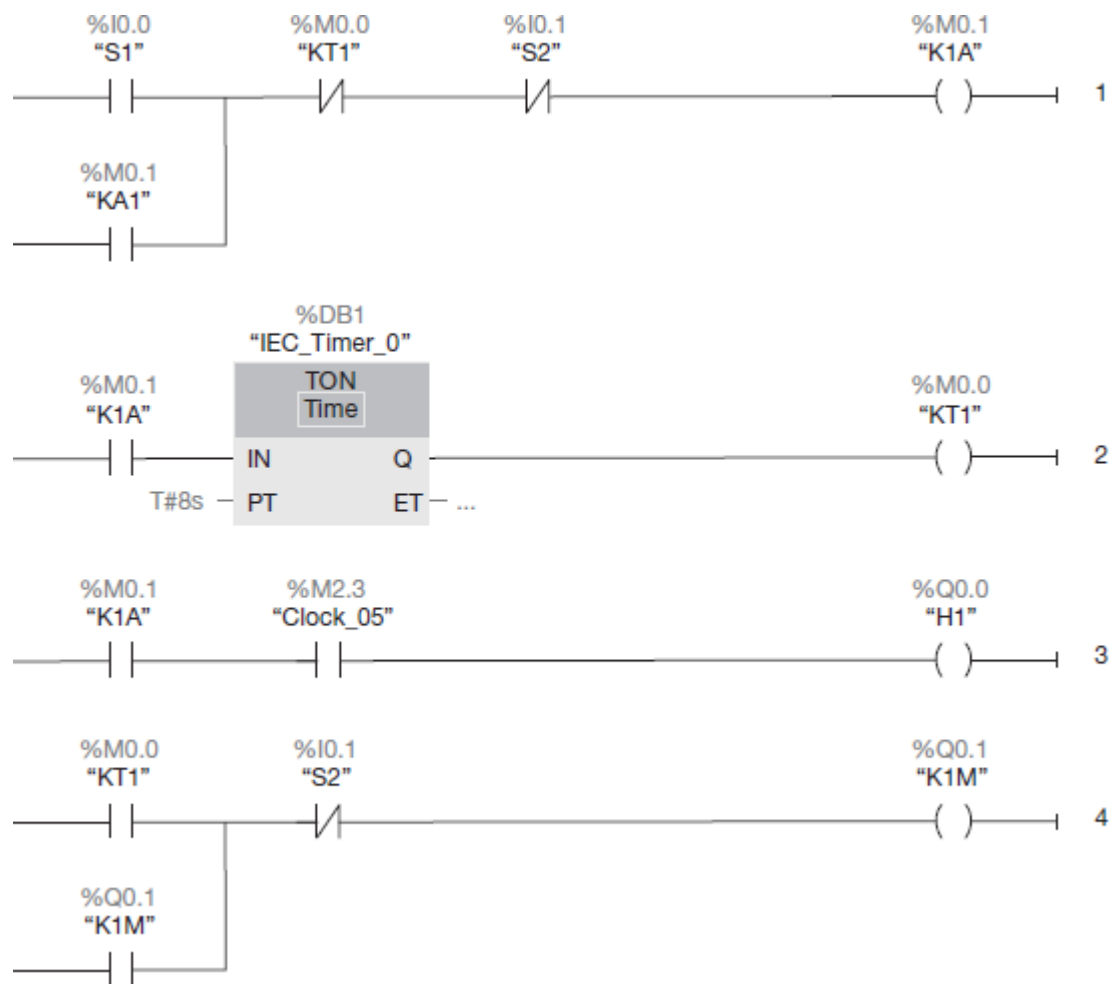


FIGURA 9.31

O funcionamento é o seguinte:

1. Na primeira linha do programa, ativa-se o relé auxiliar K1A por meio do botão S1.
2. Na segunda linha do programa, ativa-se a contagem do tempo do temporizador KT1.
3. Na terceira linha do programa, com o fechamento do contato K1A do relé auxiliar, inicia-se o lampejo da lâmpada H1 com intervalo de 0,5 s on/0,5 s off.
4. Na quarta linha do programa ocorre o fechamento do contato auxiliar do temporizador KT1 depois de 8 s. Tem lugar a energização da bobina K1M, e assim se inicia a partida da esteira transportadora. A bobina do relé auxiliar K1A é desenergizada por meio do contato KT1 do temporizador na primeira linha do programa; uma vez terminados os 8 segundos, também são desligados o temporizador e a lâmpada H1.

CONTADORES

CONTADORES

- 10.0 Generalidades
 - 10.1 Contador Crescente/Decrescente
 - 10.2 Ativação de um Contador
 - 10.3 Operação de Contagem na CPU S7-1200
 - 10.4 Aplicação: Enchimento de uma Caixa de Cerveja
-

10.0 Generalidades

Os contadores do PLC têm um formato similar ao do timer. A entrada fornece o pulso de contagem que o PLC analisa; no entanto, a outra entrada permite ressetar tudo a qualquer momento.

São disponíveis contadores **crescentes** (*counter up*) e **decrescentes** (*counter down*). Em geral, o contador crescente conta do número 0 até o valor preestabelecido, no qual depois deve acontecer uma ação. O contador decrescente parte na ordem regressiva do número preestabelecido até 0. Os contadores, na prática, efetuam contagens de eventos de todos os tipos a partir de um sensor ou transdutor que converte eventos em pulsos elétricos.

10.1 Contador Crescente/Decrescente

O exemplo da Figura 10.1 mostra um contador crescente, que aumenta progressivamente o próprio valor sempre que a entrada I0.0 passa do estado aberto (off) ao estado fechado (on). O contador não conta quando a entrada I0.1 (reset) passa do estado aberto (off) ao

CONTADORES

- 10.0 Generalidades
- 10.1 Contador Crescente/Decrescente
- 10.2 Ativação de um Contador
- 10.3 Operação de Contagem na CPU S7-1200
- 10.4 Aplicação: Enchimento de uma Caixa de Cerveja

10.0 Generalidades

Os contadores do PLC têm um formato similar ao do timer. A entrada fornece o pulso de contagem que o PLC analisa; no entanto, a outra entrada permite ressetar tudo a qualquer momento.

São disponíveis contadores **crescentes** (*counter up*) e **decrescentes** (*counter down*). Em geral, o contador crescente conta do número 0 até o valor preestabelecido, no qual depois deve acontecer uma ação. O contador decrescente parte na ordem regressiva do número preestabelecido até 0. Os contadores, na prática, efetuam contagens de eventos de todos os tipos a partir de um sensor ou transdutor que converte eventos em pulsos elétricos.

10.1 Contador Crescente/Decrescente

O exemplo da Figura 10.1 mostra um contador crescente, que aumenta progressivamente o próprio valor sempre que a entrada I0.0 passa do estado aberto (off) ao estado fechado (on). O contador não conta quando a entrada I0.1 (reset) passa do estado aberto (off) ao estado fechado (on). Na Figura 10.1, o contador é determinado no valor de 4. Então, partindo do valor 0, o contador aumenta de uma unidade o próprio valor a cada pulso que chega na entrada I0.0. Quando o PLC chega ao valor igual a 4, o bit de contagem correspondente passa do estado off ao estado on, ou seja, o contador se habilita. Um contador decrescente é similar àquele que conta de modo crescente, apenas vai diminuindo o próprio valor de contagem.

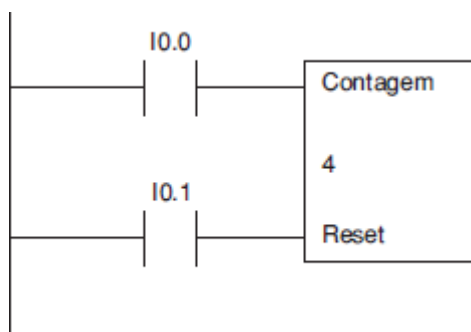


FIGURA 10.1

10.2 Ativação de um Contador

No exemplo da Figura 10.2, depois de certo número de pulsos (10) presente sobre a entrada I0.0 (fotocélula), o contato do contador crescente C1 se fecha e a saída Q0.0 se ativa. A saída poderia ser utilizada para ativar uma sinalização óptica, visando indicar a passagem de um certo número de peças sobre uma esteira. A entrada I0.1 resseta tudo em qualquer momento.

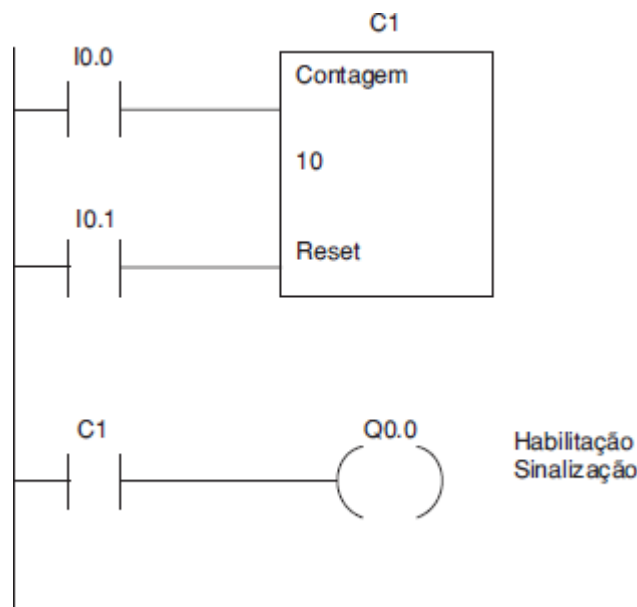


FIGURA 10.2

10.3 Operação de Contagem na CPU S7-1200

A CPU S7-1200 dispõe de três operações com contagens diferentes:

- contagem crescente (CTU)
- contagem decrescente (CTD)
- contagem crescente/decrescente (CTUD).

Como já mencionamos, cada vez que se verifica uma transição de um contato de um valor off a um valor on na entrada de contagem o contador crescente conta de modo crescente. Já o contador decrescente conta partindo de um valor predefinido quando chega à borda do sinal na entrada de contagem. O contador crescente/decrescente conta quando chega à borda do sinal na entrada de contagem. A entrada de reset resseta a contagem em qualquer momento. Os contadores da CPU S7-1200 possuem um campo numérico de contagem conforme o tipo de dados selecionados.

10.3.1 Contadores CTU, CTD, CTUD

– Contador CTU

A operação de contagem crescente CTU (veja a Figura 10.3) conta até o valor máximo quando chega à borda de subida do sinal na entrada de contagem crescente (CU). Quando o valor corrente é maior ou igual ao valor de préimposição (PV), a saída Q é ativada. Quando se ativa a entrada de reset (R), a contagem para a qualquer momento. O contador para também quando atinge o valor máximo do campo numérico selecionado.

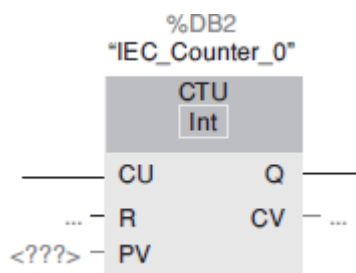


FIGURA 10.3

– Contador CTD

A operação de contagem decrescente CTD (veja a Figura 10.4) conta partindo de um valor preestabelecido PV até 0, quando chega à borda de subida do sinal, na entrada de contagem decrescente (CD). Quando o valor corrente torna-se 0, a saída Q é ativada. Com um pulso na entrada de carregamento (LOAD), o contador carrega o valor corrente com o valor preestabelecido. O contador para também quando a contagem regressiva parte do valor do campo numérico selecionado até zero.

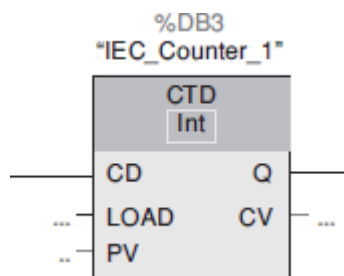


FIGURA 10.4

– Contador CTUD

A operação de contagem crescente/decrescente CTUD (veja a Figura 10.5) conta de modo crescente até um valor máximo, quando chega à borda de subida do sinal, na entrada de contagem crescente (CU), e conta de modo decrescente quando chega à borda de subida do sinal, na entrada de contagem decrescente (CD).

Quando o valor corrente é maior ou igual ao valor de pré-imposição PV, a saída QU é ativada. A entrada de reset reseta tudo em qualquer momento. Com um pulso na entrada de carregamento (LOAD), o contador carrega o valor corrente com o valor preestabelecido (PV).

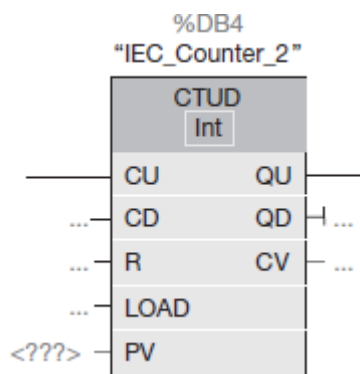


FIGURA 10.5

A Tabela 10.1 apresenta o significado dos parâmetros dos contadores.

TABELA 10.1 Tabela dos Símbolos

Parâmetro	Tipos de dados	Comentário
CU, CD	Bool	Contagem crescente e decrescente
R (CTU, CTUD)	Bool	Zera a contagem
LOAD (CTU, CTUD)	Bool	Carrega o valor preestabelecido (PV)
PV	SInt, Int, DInt, USInt, UInt, UDIInt	Valor de contagem preestabelecido
QU	Bool	Verdadeira se $CV \geq PV$
QD	Bool	Verdadeira se $CV \leq 0$
CV	SInt, Int, DInt, USInt, UInt, UDIInt	Verdadeira se $CV \leq 0$

10.3.2 O Campo Numérico de Contagem

O campo numérico de contagem dos contadores da CPU S7-1200 varia em função dos tipos de dados selecionados. Se o valor é um número inteiro sem sinal, é possível contar de forma decrescente até zero ou crescente até o valor máximo possível do campo selecionado. Se o valor é um número inteiro com sinal, é possível contar de forma decrescente até o máximo valor possível do campo selecionado positivo pela contagem crescente e negativo pela contagem decrescente.

Pela escolha do dado selecionado, clicar à direita com o mouse, até aparecer a janelinha conforme a Figura 10.6.

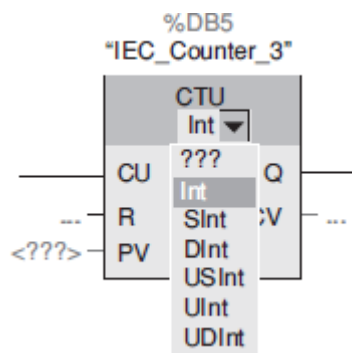


FIGURA 10.6

Escolhendo-se, por exemplo, o campo numérico SInt, conforme a Figura 10.7:

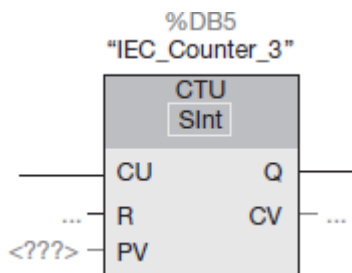


FIGURA 10.7

O valor da contagem varia de -128 a $+127$. Assim, escolhendo o campo numérico tipo UDInt, o valor da contagem varia de 0 a $+4294967295$, conforme Tabelas 6.1 e 6.2 do Capítulo 6. Na CPU S7-1200, a escolha do dado selecionado para qualquer outra instrução é efetuada da mesma forma.

10.3.3 Exemplo de Uso do Contador CTUD

Na Figura 10.8 temos um exemplo de contador CTUD, na linguagem Ladder.

A cada pulso na entrada I0.3, o contador conta de forma crescente até o valor imposto PV = 15. Após, temos a energização da saída QU, com consequente ativação da bobina Merker M0.3.

O fechamento do contato M0.3 energiza a saída discreta Q0.6.

Ao contrário, a cada pulso na entrada I0.4 o contador conta de forma decrescente até zerar.

A entrada de reset I0.5 resseta tudo a qualquer momento.

Na Figura 10.9 temos o diagrama FBD do diagrama Ladder da Figura 10.8.

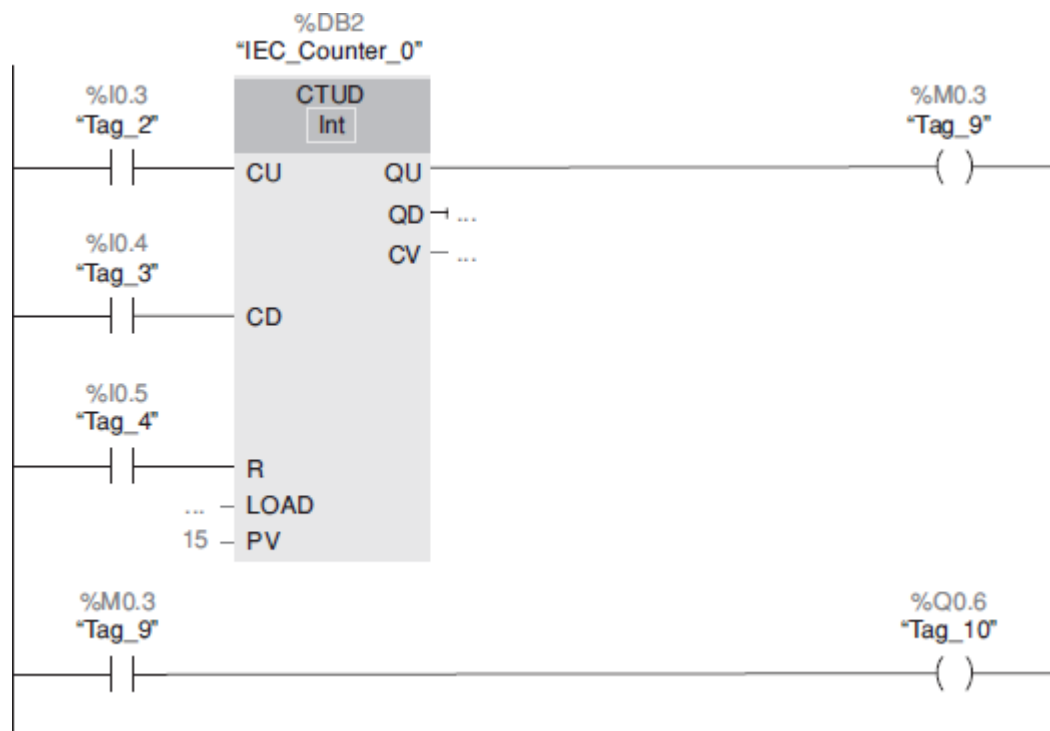


FIGURA 10.8

O diagrama Ladder resolutivo é apresentado na Figura 10.11.

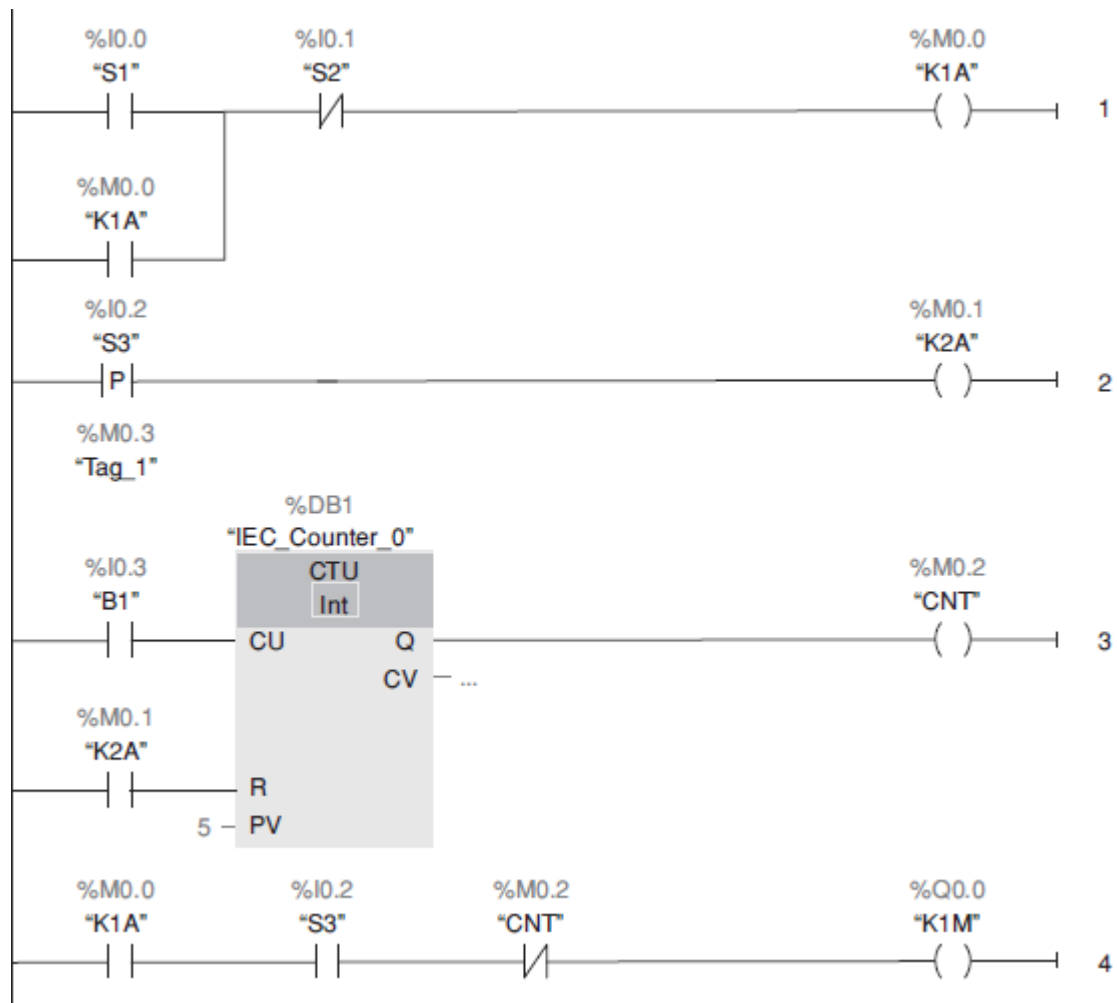


FIGURA 10.11

O diagrama FBD equivalente da Figura 10.11 é apresentado nas Figuras 10.12A e 10.12B.

O funcionamento do diagrama Ladder, em linhas gerais, é o seguinte:

1. Na linha de programa 1, quando se pressiona o botão S1 se ativa o relé auxiliar K1A, que ativa o sistema.
2. Na linha de programa 2, quando o operador põe a caixa vazia no lugar da caixa cheia, fecha-se o microinterruptor de presença da caixa S3. Parte assim um pulso por meio do contato P, a transição positiva que energiza o relé auxiliar K2A.
3. Na linha de programa 3, o fechamento do contato K2A resseta a contagem. B1 é o contato fotocélula que conta a garrafa de cerveja.
4. Na linha de programa 4, estando já fechados o contato K1A e o microinterruptor de presença da caixa S3, ativa-se o contator K1M e a esteira entra em movimento até o contador contar cinco garrafas de cerveja. Depois o contato CNT se abre, desligando o contator K1M, e a esteira para. O botão S2 para a esteira em qualquer momento.

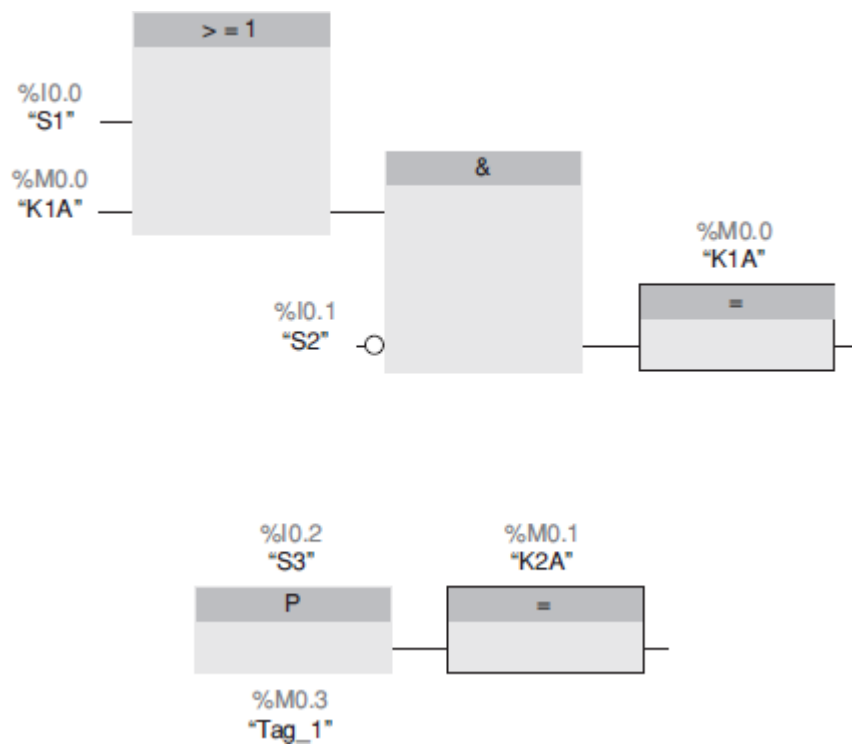


FIGURA 10.12A

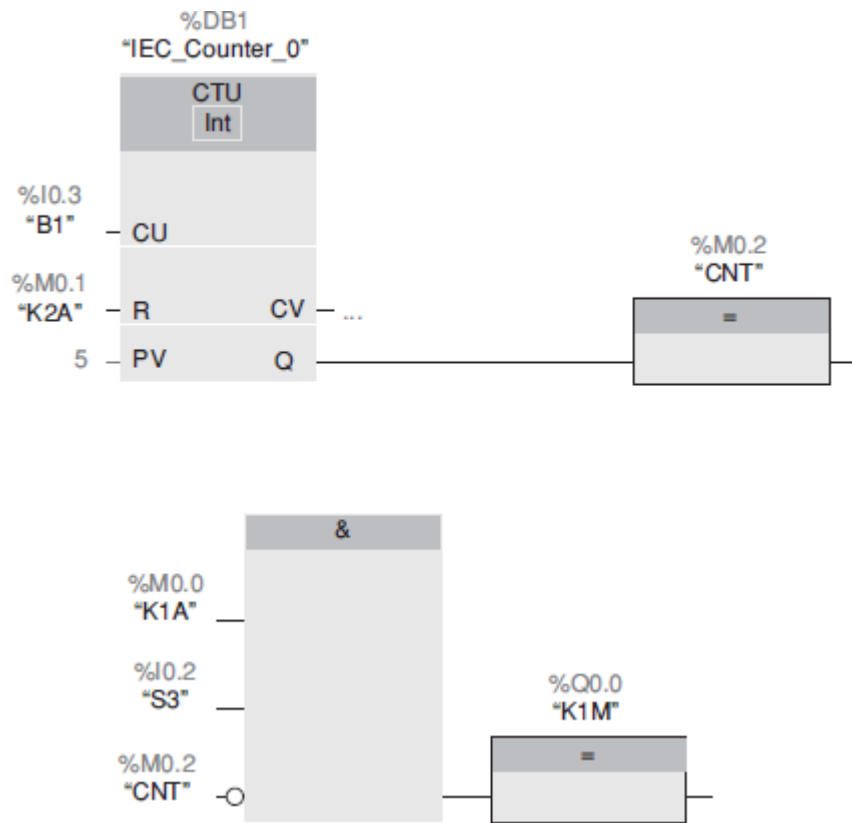


FIGURA 10.12B

APLICAÇÕES PRÁTICAS

APLICAÇÕES PRÁTICAS

- 11.0 Generalidades
 - 11.1 Cablagem do PLC S7-1200
 - 11.2 Consideração sobre as Entradas do PLC
 - 11.3 Aplicação: Escada Rolante
 - 11.4 Aplicação: Linha de Embalagem Automática
-

11.0 Generalidades

No presente capítulo será descrita uma série de aplicações práticas utilizando o PLC S7-1200. Essas aplicações têm validade para qualquer tipo de PLC. Relembremos o que foi dito nos capítulos anteriores em relação à fase principal da programação do PLC:

1. Estudar o sistema automático e representá-lo indicando o esquema de potência e o esquema funcional;
2. Nomear as entradas e as saídas do PLC I/O;
3. Codificar e escrever o programa na linguagem de programação escolhida;
4. Transferir do PC ao PLC o programa escrito;
5. Efetuar uma simulação on-line;
6. Armazenar o programa definitivamente em pen drive ou memory card;
7. Cablagem do PLC no quadro elétrico.

Neste capítulo serão discutidas as aplicações práticas, tendo presentes as fases 1, 2, 3 e 7. As fases 4, 5 e 6 já foram discutidas nos capítulos anteriores, o que será aprofundado com a utilização do software STEP 7 Basic.

APLICAÇÕES PRÁTICAS

11.0 Generalidades

11.1 Cablagem do PLC S7-1200

11.2 Consideração sobre as Entradas do PLC

11.3 Aplicação: Escada Rolante

11.4 Aplicação: Linha de Embalagem Automática

11.0 Generalidades

No presente capítulo será descrita uma série de aplicações práticas utilizando o PLC S7-1200. Essas aplicações têm validade para qualquer tipo de PLC. Relembremos o que foi dito nos capítulos anteriores em relação à fase principal da programação do PLC:

1. Estudar o sistema automático e representá-lo indicando o esquema de potência e o esquema funcional;
2. Nomear as entradas e as saídas do PLC I/O;
3. Codificar e escrever o programa na linguagem de programação escolhida;
4. Transferir do PC ao PLC o programa escrito;
5. Efetuar uma simulação on-line;
6. Armazenar o programa definitivamente em pen drive ou memory card;
7. Cablagem do PLC no quadro elétrico.

Neste capítulo serão discutidas as aplicações práticas, tendo presentes as fases 1, 2, 3 e 7. As fases 4, 5 e 6 já foram discutidas nos capítulos anteriores, o que será aprofundado com a utilização do software STEP 7 Basic.

11.1 Cablagem do PLC S7-1200

A Figura 11.1 fornece informações relativas à cablagem da CPU 1212C com alimentação a 230 VAC e saída AC/DC a relé.

Na Figura 11.1 estão indicadas as cablagens elétricas de entrada e de saída. As entradas são indicadas genericamente com interruptores abertos. Nas entradas podemos conectar botões, chave de posição e vários sensores. As saídas são indicadas geralmente com bobinas, e podemos conectar nelas os atuadores, como lâmpadas, eletroválvulas, bobinas e outros.

A alimentação das entradas é efetuada com uma fonte de alimentação interna de 24 VDC.

A alimentação da saída é comum, ou seja, tem a mesma tensão de alimentação para todas as saídas. As saídas com alimentação separada também podem ser conectadas. No esquema da Figura 11.1, para fins de simplificação, não estão indicados dispositivos de proteção, tais como fusíveis e disjuntor termomagnético. É notório que o manual de sistemas deve ser consultado em cada caso.

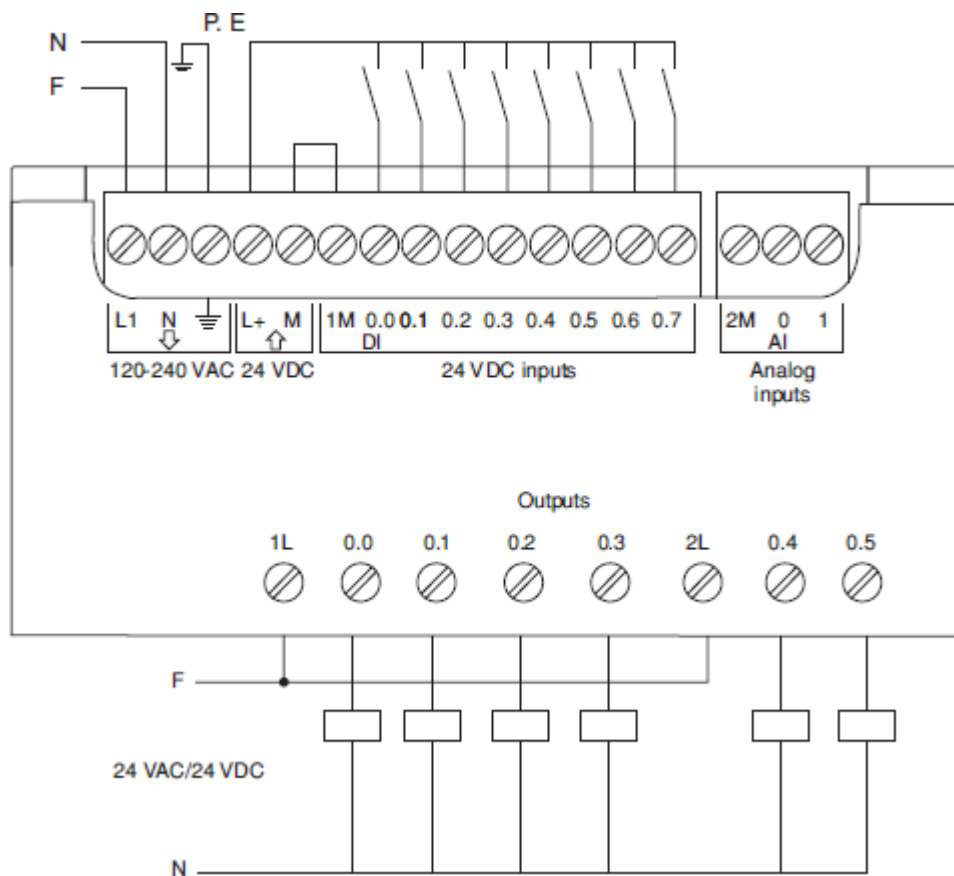


FIGURA 11.1

A CPU 1212C é dotada no módulo básico de 8DI/6DO, ou seja, oito entradas digitais e seis saídas digitais. As entradas são indicadas via software com a letra %I e as saídas, com a letra %Q. A CPU1212C tem entradas que vão de %I0.0 a %I0.7, e as saídas vão de %Q0.0 a %Q0.5.

11.2 Consideração sobre as Entradas do PLC

Propomos uma observação sobre as entradas do PLC na cablagem e sobre a lógica do programa.

- Se na lógica do programa uma entrada como, por exemplo, I0.1 (S2) é apresentada com o símbolo , quando for cabeada deverá ser conectada como uma entrada **normalmente aberta NA**. Veja a Figura 11.2.

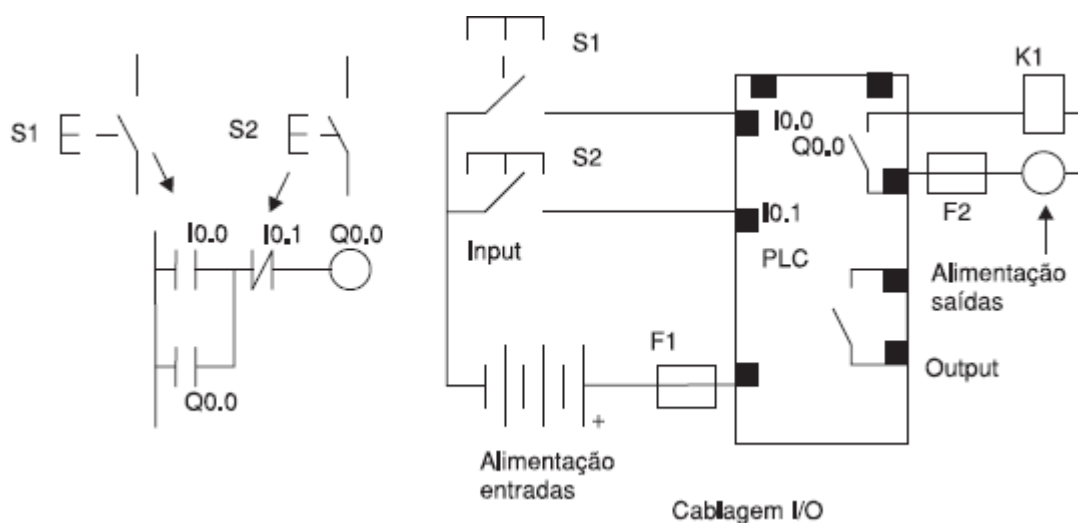


FIGURA 11.2

- Se na lógica do programa uma entrada como, por exemplo, I0.1 (S2) é apresentada com o símbolo , quando for cabeada deverá ser conectada como uma entrada **normalmente fechada NF**. Veja a Figura 11.3. Isso acontece porque a lógica do PLC trabalha geralmente em lógica positiva.

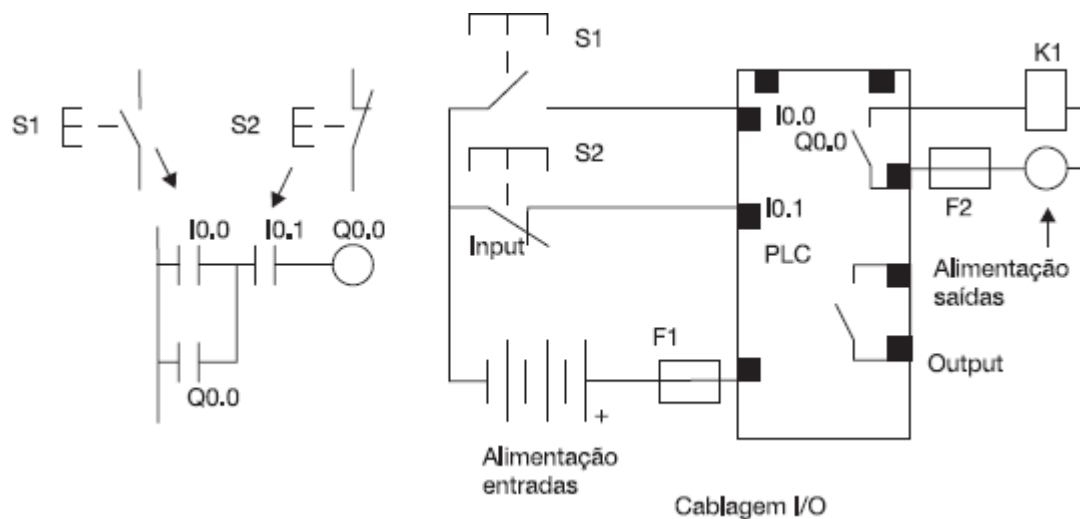


FIGURA 11.3

11.3 Aplicação: Escada Rolante

Esta aplicação de escada rolante é muito difundida nas grandes cidades. Encontra-se com mais frequência nos metrô e *shoppings*. A escada rolante se aciona automaticamente quando uma pessoa se aproxima do início da mesma. A pessoa então é transportada até o ponto final da escada. Se no tempo de 10 segundos outras pessoas sobem na escada rolante, a escada continua a funcionar. Se depois de 10 segundos ninguém subir na escada, ela para automaticamente.

Ou seja, ela só funciona se tem alguém a utilizando; do contrário, para. Essa aplicação surgiu para economizar energia no período em que não tem ninguém circulando na escada. A Figura 11.4 exemplifica essa aplicação.

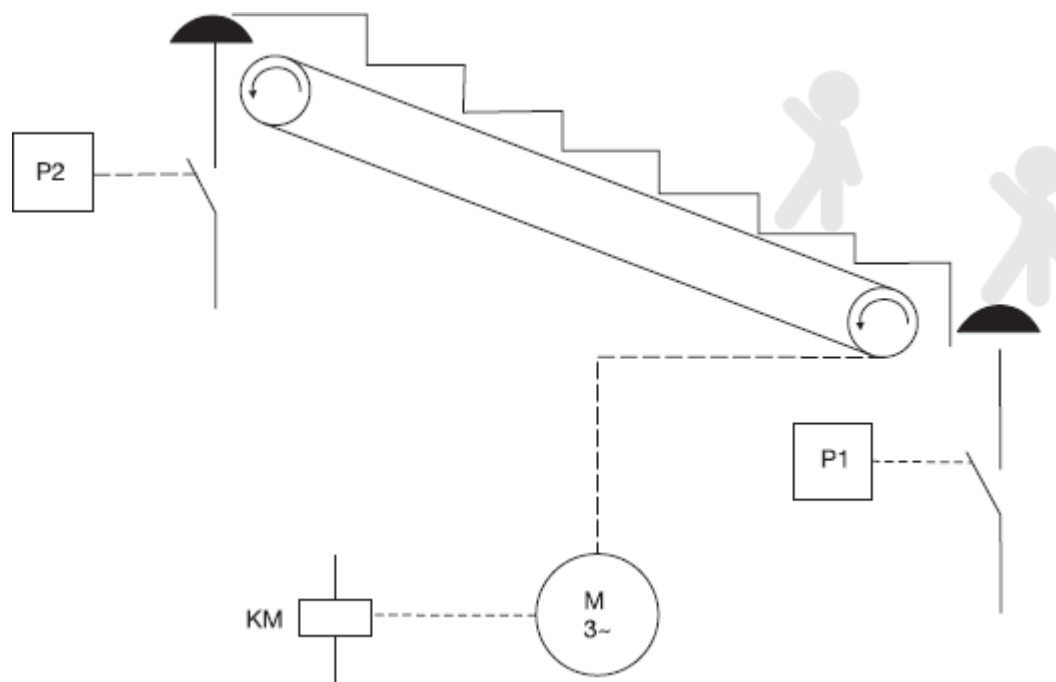


FIGURA 11.4

Na Figura 11.6 está apresentado o esquema elétrico funcional da escada rolante.

Notamos na Figura 11.6 que a escada rolante se aciona somente quando uma pessoa pisa no sensor de peso P1 situado no início da escada. A chave P1 se fecha, energizando assim a bobina do relé auxiliar K1A.

A mesma chave auxiliar K1A (5), ao se fechar, aciona ainda o contator KM; assim a escada rolante começa a funcionar. Quando chega ao fim da escada rolante, a pessoa pisará no sensor de peso P2, colocado no final da escada, desenergizando assim a bobina do relé auxiliar K1A e a bobina do timer KT1. Nesse mesmo instante a bobina de KT1 iniciará a contagem do tempo de 10 segundos, e o motor da escada rolante continua a funcionar por meio do contato timer KT1 (4), que está fechado, alimentando assim a bobina KM.

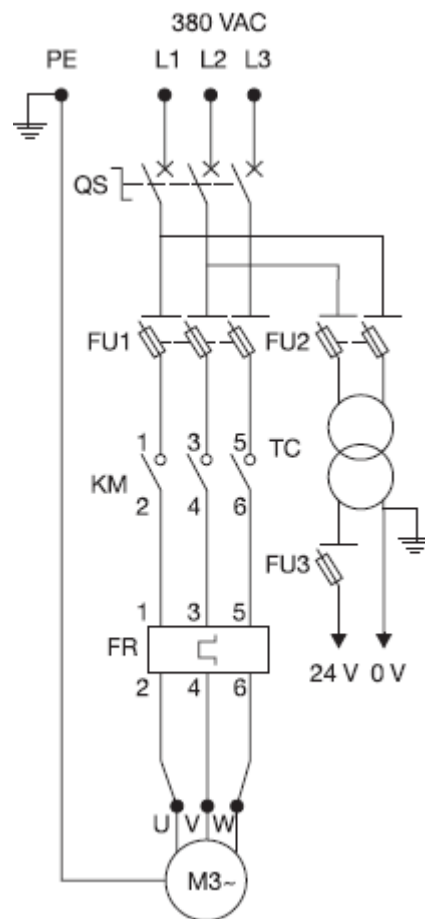


FIGURA 11.5 Diagrama de potência.

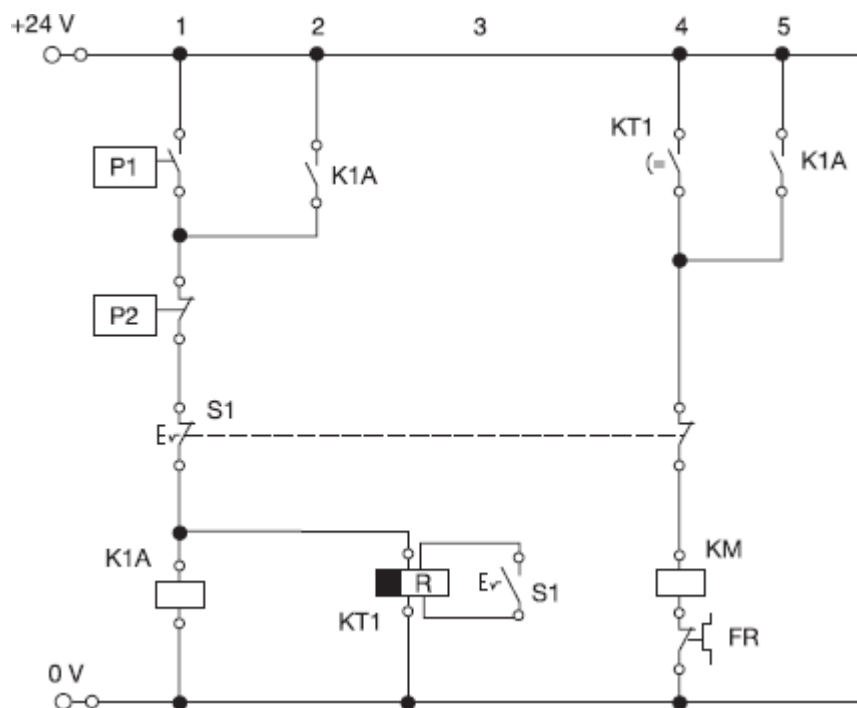


FIGURA 11.6

Passados os 10 segundos, o contato KT1 do timer se abre, desenergizando a bobina do contator KM, e a escada rolante para.

Devemos, porém prever o que acontece se no intervalo de 10 segundos de contagem do timer uma outra pessoa pisa no sensor de peso P1. Nesse caso, a chave P1 se fecha novamente, energizando outra vez a bobina do relé auxiliar K1A. Acontagem do tempo é então interrompida, permitindo que a escada rolante continue a funcionar. O botão de stop S1 para a escada rolante a qualquer momento, ressetando a contagem do timer.

Tabela 11.1 Tabela dos Símbolos

Símbolos	Endereço	Comentários
P1	I0.0	Sensor peso início escada

P2	I0.1	Sensor peso fim escada
S1	I0.2	Botão stop e reset timer
FR	I0.3	Relé térmico motor
KM	Q0.0	Contator motor escada rolante
K1A	M0.0	Merker de relé auxiliar
KT1	M0.1	Saída timer com atraso no desligamento

Na Figura 11.7 apresenta-se o esquema Ladder resolutivo da Figura 11.6.

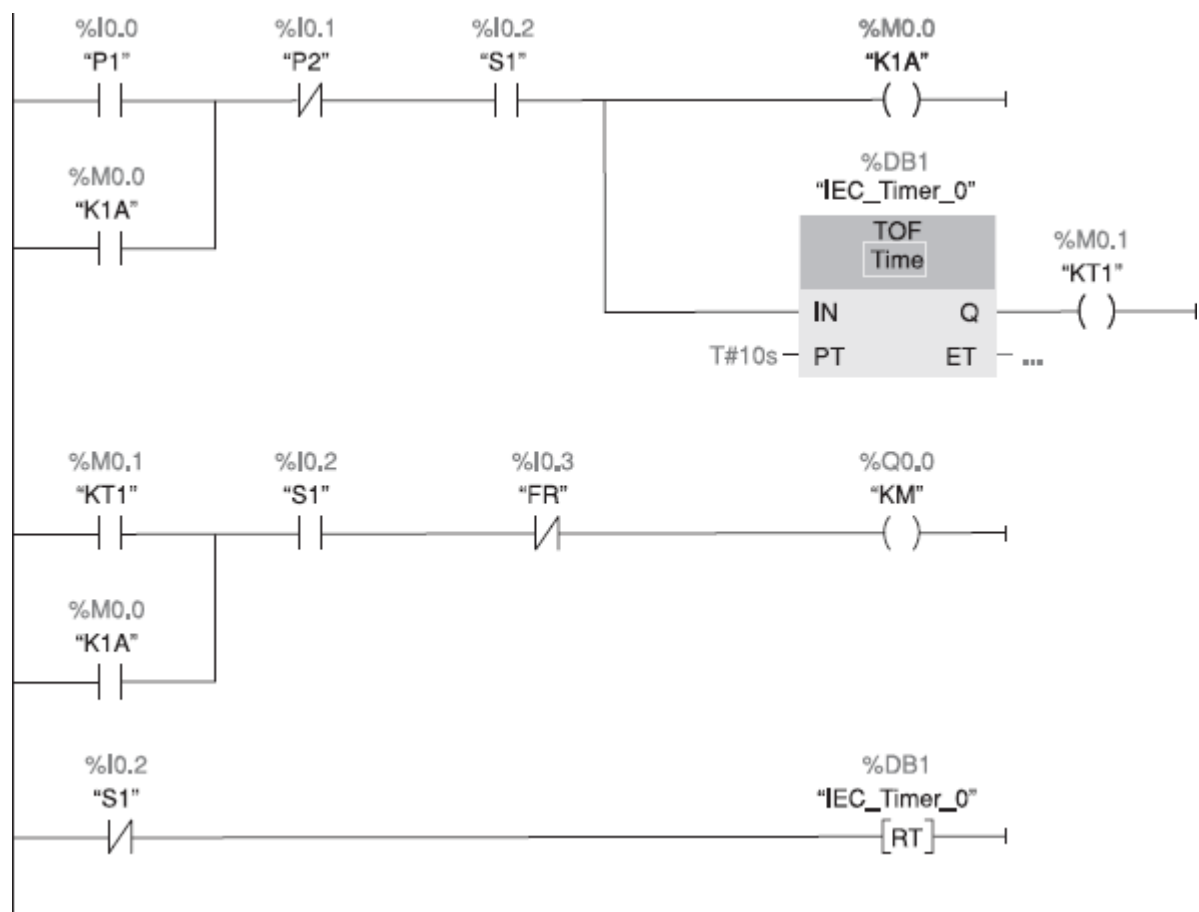


FIGURA 11.7

Na Figura 11.8 apresenta-se o diagrama FBD equivalente da Figura 11.7.

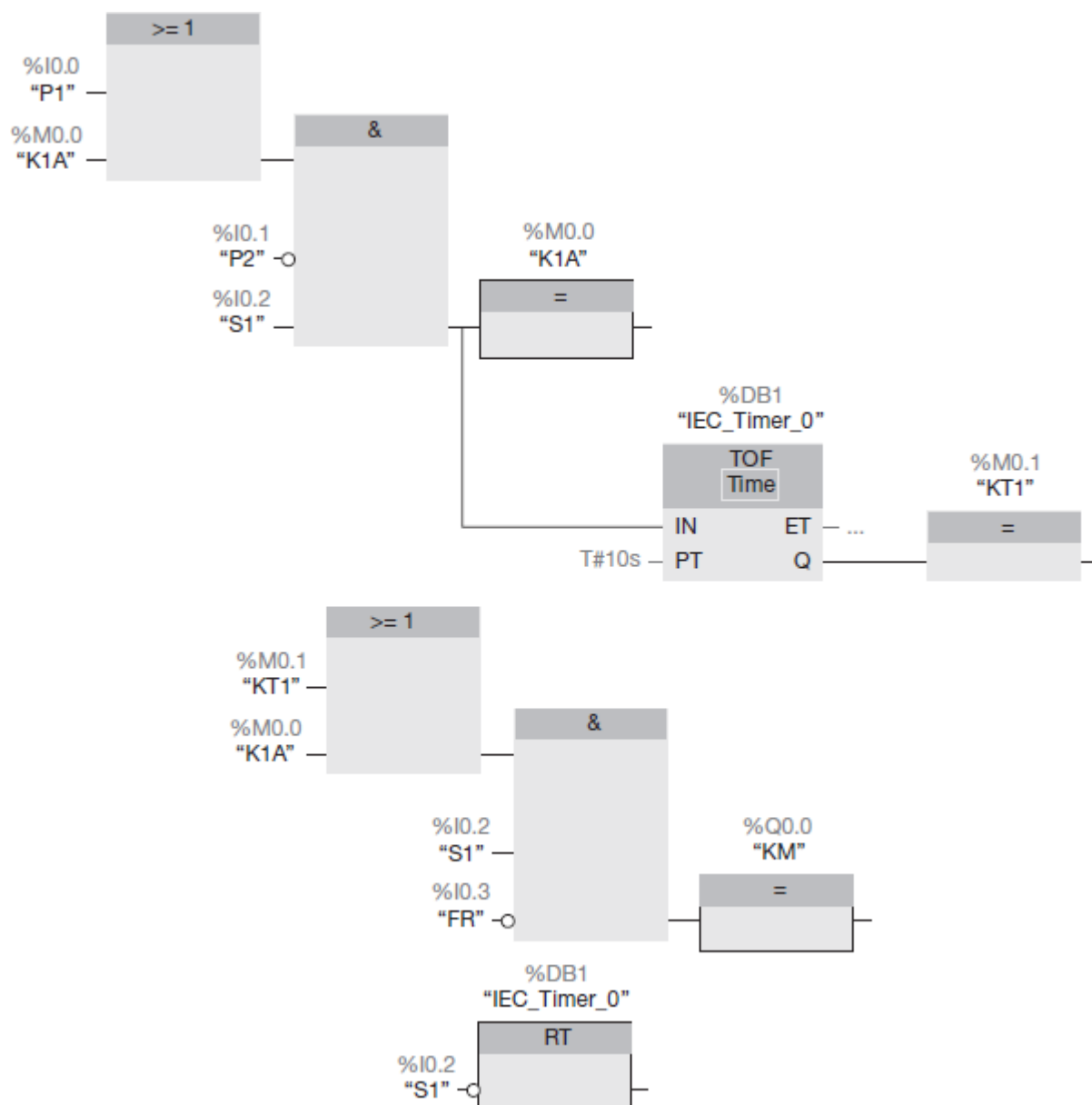


FIGURA 11.8

Na Figura 11.9 temos o cabeamento completo da CPU 1212C da escada rolante. Na Figura 11.9 pode-se notar a presença do relé térmico FR para proteção do motor trifásico contra sobrecarga. Vê-se nesse caso que a intervenção do térmico FR desenergiza, via hardware, o contator KM, mediante o seu contato normalmente fechado em série à bobina KM, e assinala a sua intervenção ao PLC, que desabilita a saída Q0.0 por meio do contato Hardware normalmente aberto (NA) ligado na entrada I0.3. Essa solução, evidentemente, é mais custosa, porém é a mais segura, porque permite salvaguardar o motor também em caso de defeito do PLC ou de defeito de um de seus I/O. De fato, o contator KM é desenergizado por meio do contato Hardware normalmente fechado do relé térmico FR.

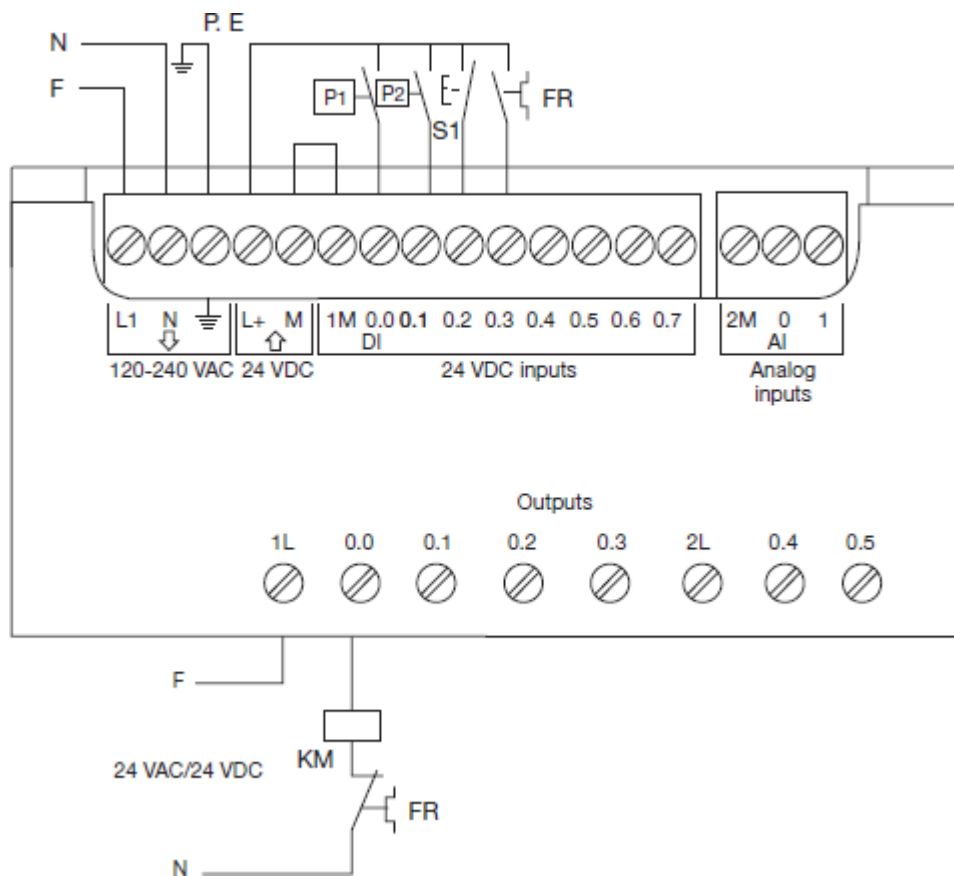


FIGURA 11.9

11.4 Aplicação: Linha de Embalagem Automática

A embalagem dos produtos é uma das mais importantes no setor da automação industrial, e pode ser resolvida com o uso de controladores programáveis.

A Figura 11.10 ilustra um simples sistema de embalagem para o enchimento de caixas de cocos; cada caixa contém no máximo 10 cocos.

A esteira transportadora 1 carrega e transporta os cocos, enquanto a esteira 2 providencia o transporte das caixas de coco, que caem da esteira 1 até a caixa posicionada na esteira 2.

O coco que cai na caixa é contado por meio da fotocélula B1. Ao atingir o valor de 10 cocos, a esteira 1 para e a esteira 2 parte. Na Figura 11.11 é apresentado o diagrama elétrico de potência dos motores para o acionamento das esteiras.

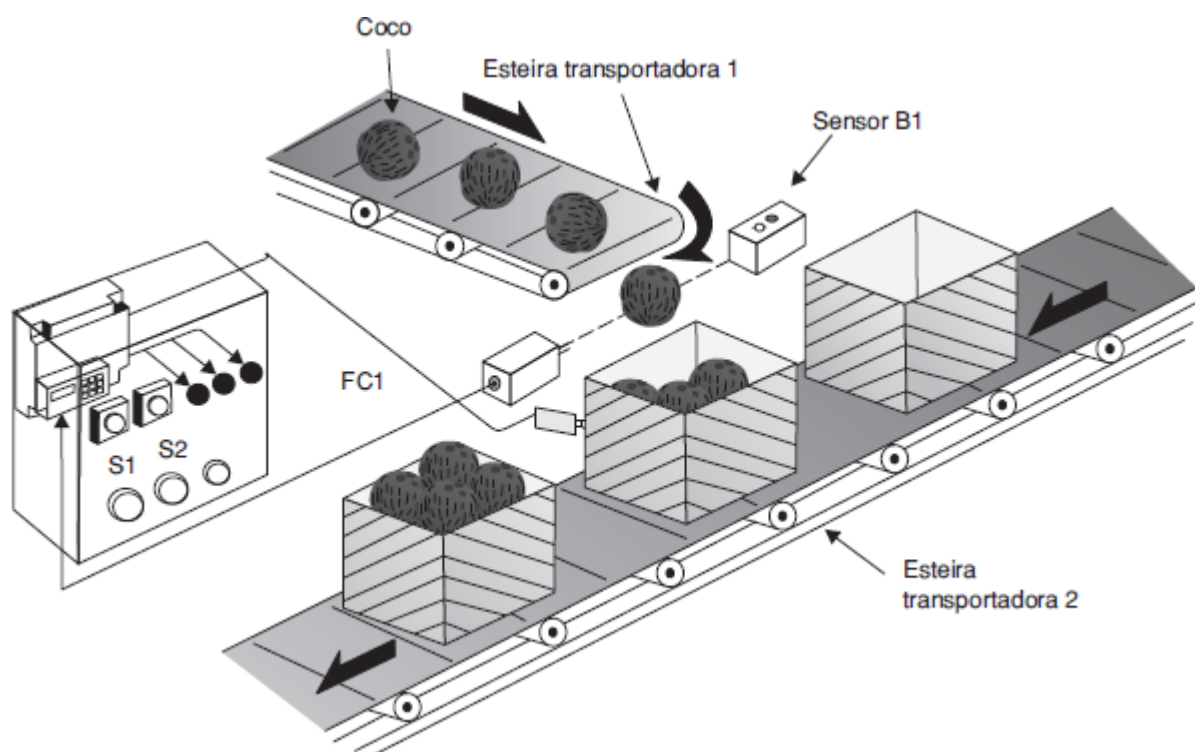


Tabela 11.2 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão início ciclo automático
S2	I0.1	Botão parada
FR1	I0.2	Térmica motor esteira 1
FR2	I0.3	Térmica motor esteira 2
B1	I0.4	Fotocélula
FC1	I0.5	Fim de curso de presença caixa
K1M	Q0.0	Contator esteira 1
K2M	Q0.1	Contator esteira 2
K1A	M0.0	Merker de relé auxiliar
KT1	M0.1	Saída temporizador
CNT	M0.2	Saída contador crescente

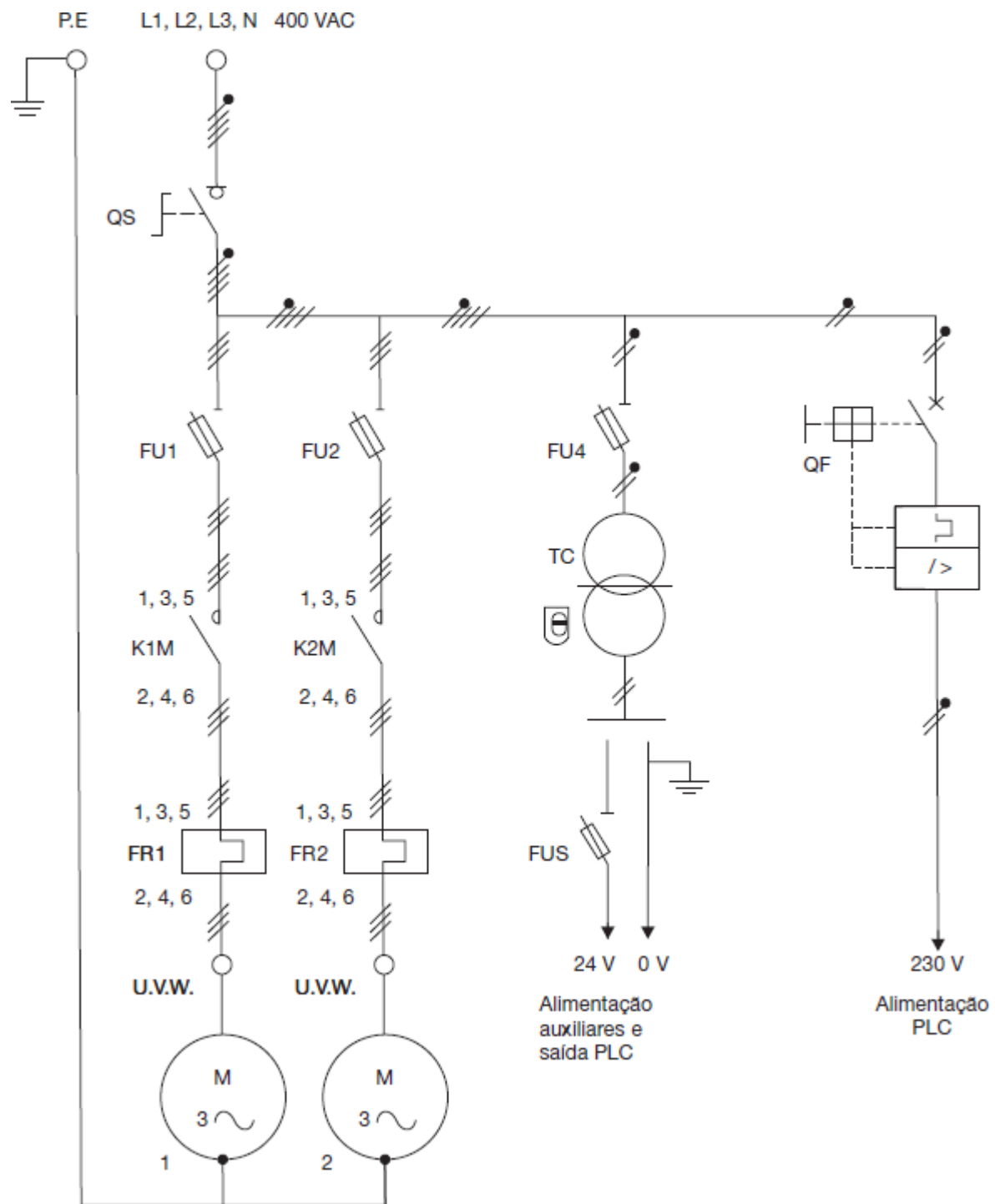


FIGURA 11.11

Nas Figuras 11.12A e 11.12B apresenta-se o esquema Ladder resolutivo da linha de embalagem automática.

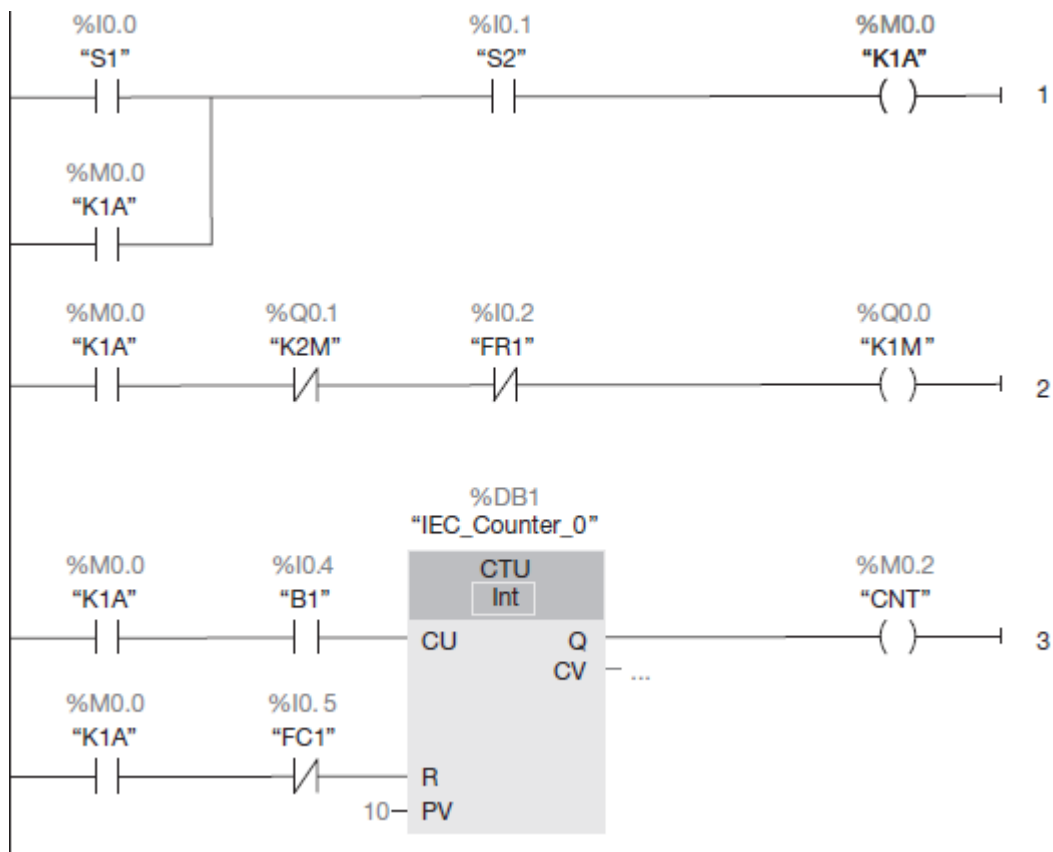


FIGURA 11.12A

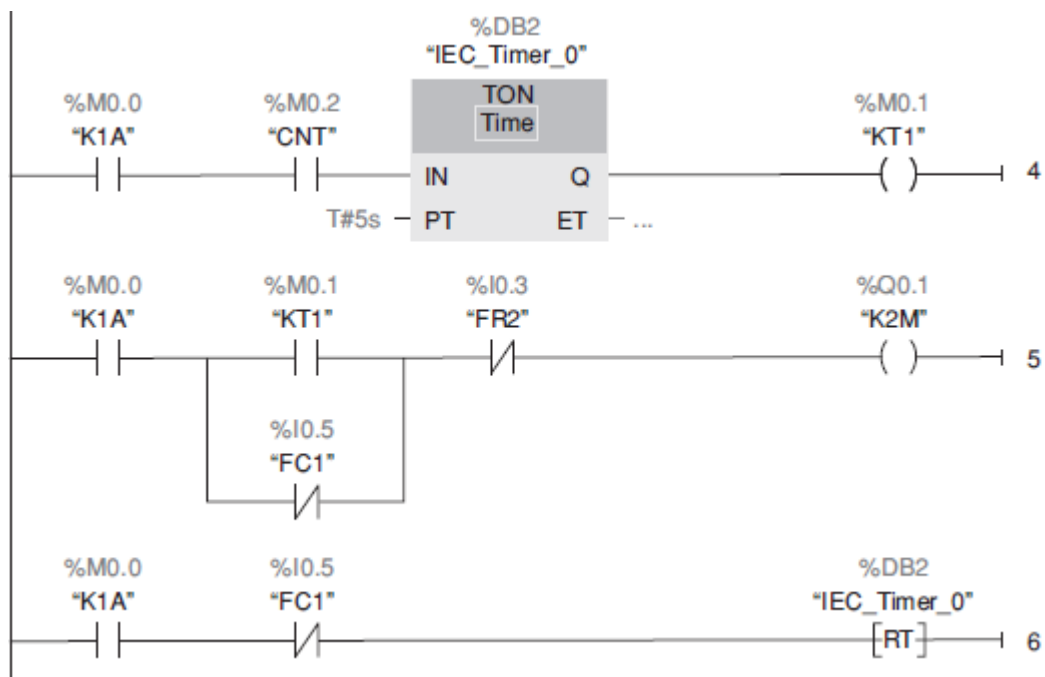


FIGURA 11.12B

- Na linha de programa 1, temos a presença do botão de start S1 e stop S2 do ciclo automático de embalagem.
- Na linha de programa 2, pressionando o botão de start S1 (veja linha 1), temos a fechadura dos contatos Merker K1A em todas as linhas do programa; por consequência, energiza-se a bobina K1M que ativa a esteira 1 que carrega e transporta o coco.
- Na linha de programa 3, o coco, ao cair um a um na caixa, é assim detectado pela fotocélula B1, que, energizando-se, ativa a contagem para a frente do contador CNT.
- Na linha de programa 4, após contagem de 10 (total dos cocos), fecha-se o contato normalmente aberto CNT, que, por sua vez, ativa a contagem do timer KT1.
- Na linha de programa 5, após 5 segundos fecha-se o contato normalmente aberto KT1, que ativa a esteira 2 por meio da bobina K2M. Nesse momento, a esteira 1 para por meio do contato normalmente fechado K2M em série com a bobina K1M (veja linha de programa 2).

O timer KT1 tem a função de criar um tempo de atraso entre a queda do décimo coco e a ativação efetiva da esteira 2. Temos assim a certeza de que todos os 10 cocos estão na caixa.

Notamos como o fim de curso FC1 de presença da caixa é normalmente fechado. Na posição, conforme a Figura 11.10, a caixa toca o fim de curso corretamente, então o FC1 é aberto.

- Na linha de programa 6, com a partida da esteira 2, o fim de curso FC1 comuta de aberto a fechado. Temos assim o reset do timer KT1 e do contador CNT.

Na linha de programa 5, o contato FC1, estando fechado, permite que a bobina K2M fique energizada, dado que a caixa não toca o fim de curso FC1 (caixa em movimento).

A esteira 2, ao se movimentar, permite o posicionamento da caixa subsequente. A caixa subsequente permite que o fim de curso FC1 comute novamente.

O contato FC1 passa assim de normalmente fechado a aberto; em consequência, a bobina K2M se desenergiza e a esteira 2 para, assim, a esteira 1 parte novamente.

O ciclo recomeça de forma automática.

Na Figura 11.13 temos o cabeamento completo da CPU 1212C da linha de embalagem automática.

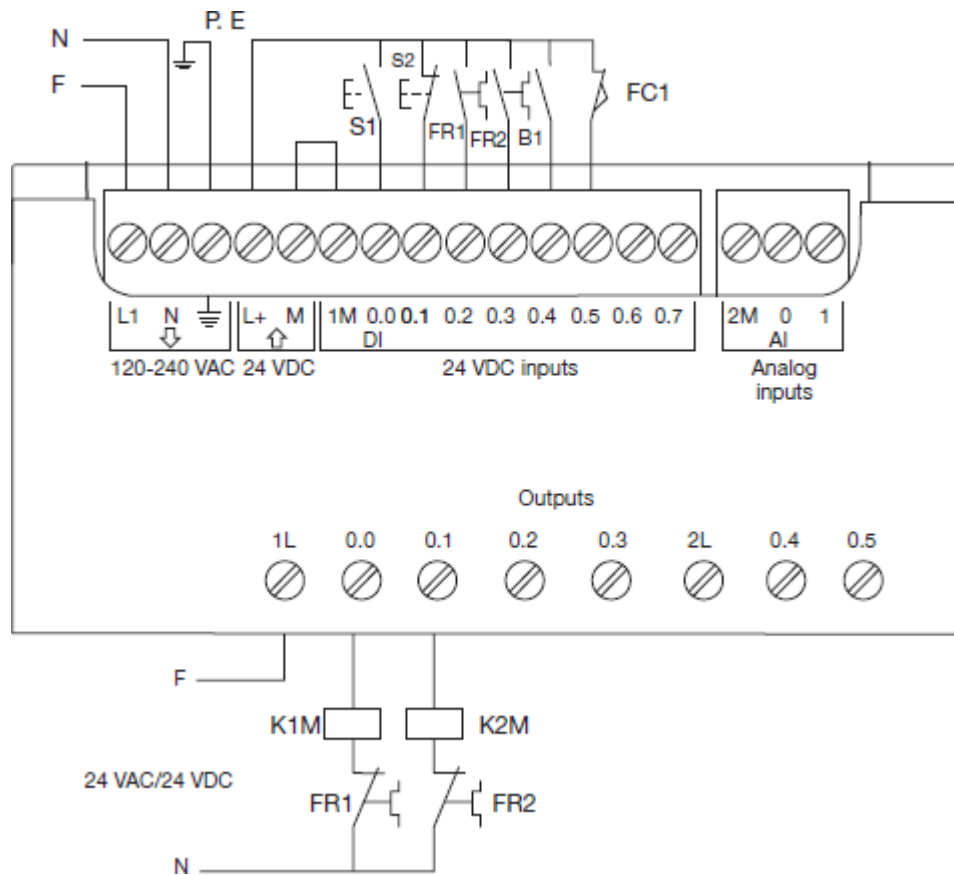


FIGURA 11.13

BERGER, H. *Automating with SIMATIC S7-1200*. 1. ed. Siemens. Nuremberg, 2011.

PRUDENTE, F. *Automação industrial: PLC-Programação e instalação*. 1. ed. Rio de Janeiro: LTC, 2010.

PRUDENTE, F. *Automação industrial: PLC-Teoria e aplicações/curso básico*. 2. ed. Rio de Janeiro: LTC, 2010.

MANUAIS

– Siemens. *Getting Started with SIMATIC S7-1200*. Nuremberg, 2009.

– Siemens. *S7-1200 Programmable Controller, system manual*. Nuremberg, 2009.